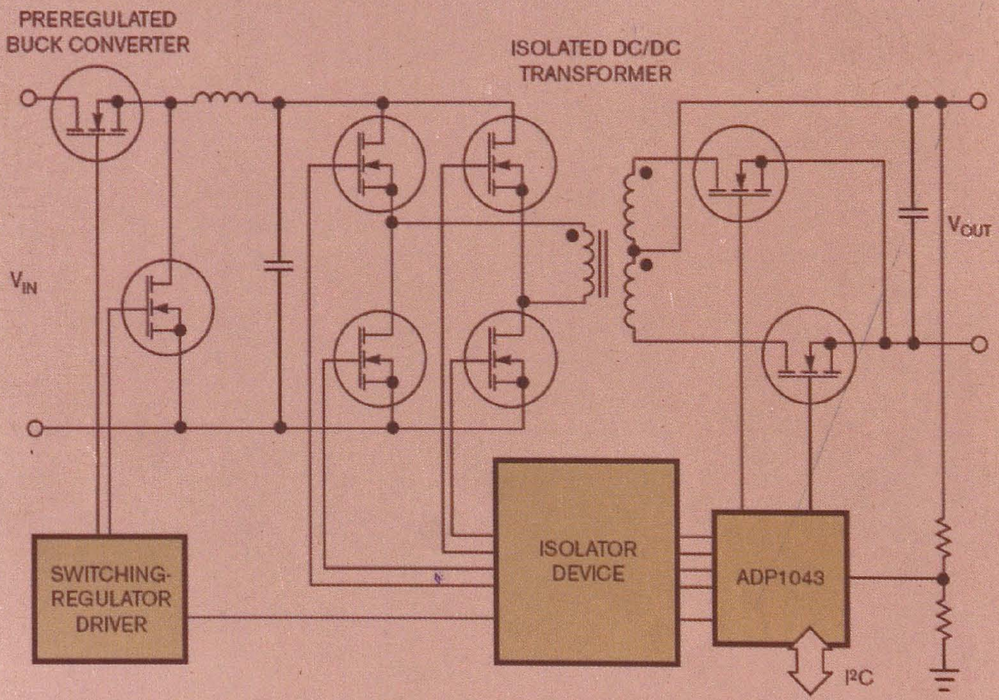
நுண்முறைப்படுத்தியும் நுண்கட்டுப்படுத்தியும்

க. தமிழ்ச்செல்வி
கு. ரேணுகா தேவி



தமிழ்நாடு மாநில உயர்கல்வி மன்றம்
காமராஜர் சாலை, சென்னை-600 005.

முதற் பதிப்பு	:	2011
பதிப்புரிமை	:	தமிழ்நாடு மாநில உயர்கல்வி மன்றம் சென்னை – 600 005
நூலின் பெயர்	:	நுண்முறைப்படுத்தியும் நுண்கட்டுப்படுத்தியும்
நூலாசிரியர்	:	முனைவர் க. தமிழ்ச்செல்வி, துணைப்பேராசிரியர், இயற்பியல் துறை, பெரியார் ஈ.வே.ரா. அரசுக் கலைக் கல்லூரி, திருச்சிராப்பள்ளி 620 001. முனைவர் கு. ரேணுகா தேவி, இணைப்பேராசிரியர், அரசு மகளிர் கலைக் கல்லூரி, புதுக்கோட்டை 622 001.
மறு ஆய்வு செய்தவர்	:	முனைவர் ஆர். சபேசன், பேராசிரியர், துறைத் தலைவர் (ஓய்வு), இயற்பியல் துறை, அழகப்பா பல்கலைக்கழகம், காரைக்குடி – 630 003
தமிழ் திருத்தம் செய்தவர்	:	முனைவர் இரா. இராமன், இணைப் பேராசிரியர், தமிழ்த்துறை, அரசினர் ஆடவர் கலைக் கல்லூரி, (தன்னாட்சி) நந்தனம், சென்னை 600 035.
விலை	:	ரூ.81.00
அச்சிட்டோர்	:	பாவை பிரிண்டர்ஸ் (பி) லிட்., 16(142), ஜானி ஜான்கான் ரோடு, இராயப்பேட்டை, சென்னை - 600 014. போன் : 28482441, 28482973

MADURAI KAMARAJ UNIVERSITY
299067

பொருளடக்கம்

வ.எண்	தலைப்பு	பக்க எண்
1	இன்டெல் 8085 நுண்முறைப்படுத்தியின் கட்டமைப்பு மற்றும் திட்டநிரல்	1
2	இடைப் பரிமாற்று நினைவகம் மற்றும் உள்ளீடு / வெளியீடு கருவிகள்	56
3	இடைப் பரிமாற்றுப் புறவியல் சாதனங்கள்	103
4	இன்டெல் 8086 நுண்முறைப்படுத்தி	152
5	8051 நுண்கட்டுப்படுத்தியின் கட்டுமானம் மற்றும் திட்டநிரல்படுத்தல்	223

1
இன்டெல் 8085 நுண்முறைப்படுத்தியின் கட்டமைப்பு மற்றும் திட்டநிரல்
(Architecture and Programme of Intel 8085 Microprocessor)

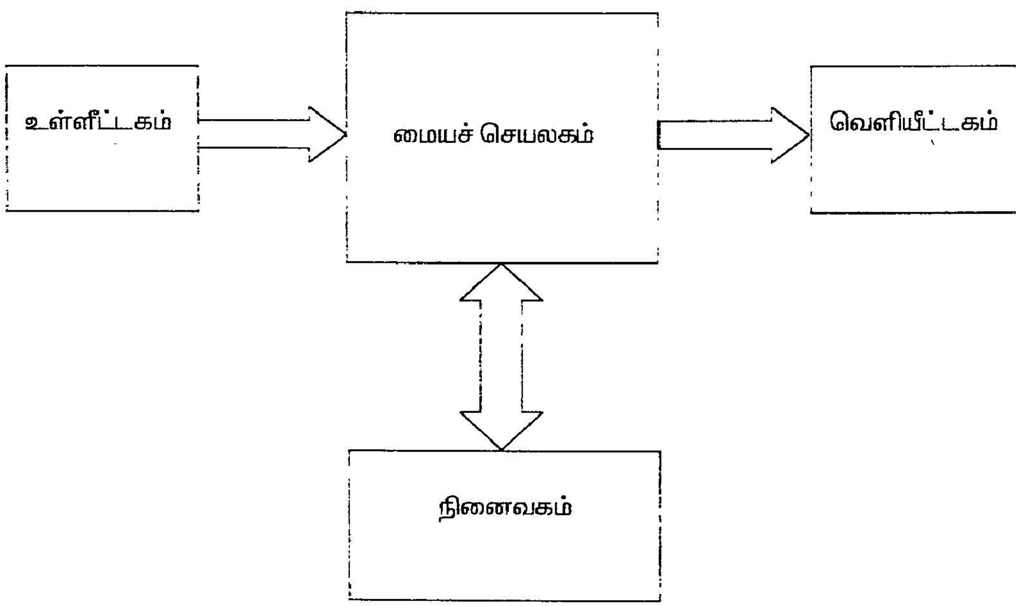
1.1 முன்னுரை

தொழில்மயமான சமுதாயத்தில் அன்றாட வேலைகளில் நுண்முறைப்படுத்தி மிக முக்கிய பங்கு வகுக்கிறது. நுண்முறைப் படுத்தியை ஒரு கணினியின் தரவுகளை முறைப்படுத்தும் சாதன மாகவோ அல்லது கணக்கீடு சாதனமாகவோ கருதலாம். இது ஒரு நுண்கணினியின் மையச் செயலகம் போன்று செயல்படும். ஒரு பெரிய அளவிலான தொகுப்பு சில்லில் (LSI Chip) வடிவமைக்கப் பட்ட மையச் செயலகமே நுண்முறைப்படுத்தியாகும். இது ஈரடி எண்களான 0 மற்றும் 1 ஐ கொண்டு இயங்கவும் மற்றும் தொடர்பு கொள்ளவும் செய்கிறது. ஒவ்வொரு நுண்முறைப்படுத்திக்கும் ஒரு நிரந்திர கட்டளைத் (Instruction) தொகுப்பு அதாவது திட்ட நிரல் (Program) உள்ளது. அவை இயந்திர மொழி (Machine language) எனப்படும் ஈரடி முறையில் இருக்கும். ஆனால் இயந்திர மொழியில் திட்ட நிரல் எழுதுவது மிகவும் கடினம். எனவே இக்குறைப் பாட்டினைத் தவிர்ப்பதற்கு ஈரடி கட்டளைகள் செயற்குறிப்பு (Mnemonics) மூலம் குறிப்பிடப்பட்டது. திட்ட நிரல், செயற்குறிப்பும் இயக்கக் குறியீடும் கொண்ட இடைநிலை மொழியில் (Assembly language) எழுதப்பட்டது.

முதல் நுண்முறைப்படுத்தி 1971ஆம் ஆண்டு அமெரிக்காவில் உள்ள இன்டெல் தொழிலகத்தால் அறிமுகம் செய்யப்பட்டது. இது இன்டெல் 4004 என்ற 4–பிட் நுண்முறைப்படுத்தியாகும். பிறகு 1972 ஆம் ஆண்டு அதே தொழிலகம் 8008 என்ற 8–பிட் நுண்முறைப்படுத்தியை அறிமுகம் செய்தது. 4–பிட் மற்றும் 8–பிட் நுண்முறைப்படுத்திகள் என்பவை அவை தரவுகளை (Data) ஒரே நேரத்தில் கையாளும் முறையை குறிப்பிடுகின்றன. தற்போது இன்டெல் 8085, மோட்டரோலா MC 6809 போன்ற 8–பிட் நுண்முறைப்படுத்திகள் பயன்பாட்டில் உள்ளன. பின்வரும் பிரிவுகளில் இன்டெல் 8085 நுண்முறைப்படுத்தியின் கட்டமைப்பு, முனை அமைப்பு மற்றும் இடைநிலை மொழியில் எழுதப்பட்ட அதனுடைய திட்ட நிரல்கள் பற்றிக் காண்போம்.

1.2 எண்ணிலக்கக் கணினி
(Digital Computer)

எண்ணிலக்கக் கணினி என்பது செயல்முறைப்படுத்தக் கூடிய கருவியாகும். இது கணக்கீடு செய்வதற்காகவே வடிவமைக்கப்பட்டதாகும். மையச் செயலகம், நினைவகம், உள்ளீடு மற்றும் வெளியீடு சாதனங்கள் ஒரு எண்ணிலக்கக் கணினியின் முக்கியமான பாகங்களாகும். ஒரு நுண்கணினி (Micro computer) என்பது சிறிய எண்ணிலக்கக் கணினியாகும். ஒரு நுண்கணினியின் மையச் செயலகமே நுண்முறைப்படுத்தியாகும். ஒரு நுண்கணினியின் அமைப்பு, படம் (1.1)இல் காட்டப்பட்டுள்ளது. ஒரு கணினியின் உள்கட்டச் சாதனங்கள் (Physical devices) மற்றும் மின்சுற்றுகள், வன்பொருள் (Hardware) எனப்படும். ஒரு குறிப்பிட்ட வேலையைச் செய்துமுடிப்பதற்காக எழுதப்படும் கட்டளைகளின் (Instructions) தொகுப்பு திட்ட நிரல் (Program) எனப்படும். ஒரு குறிப்பிட்ட கணினிக்காக எழுதப்படும் திட்ட நிரல், அந்தக் கணினியின் மென்பொருள் (Software) எனப்படும். உள்ளீடு மற்றும் வெளியீடு சாதனங்கள் ஒன்றாக இணைந்து, புறவியலான சாதனங்கள் (Peripheral devices) எனப்படும். எண்ணிலக்கக் கணினியின் முக்கியமான பாகங்களைப் பற்றிய விளக்கத்தைக் காண்போம்.



படம் (1.1) நுண்கணினியின் அமைப்பு

1.2.1. மையச் செயலகம் (Central processing unit)

இந்த அலகு கட்டளைகளை நினைவகத்திலிருந்து ஒன்றன்பின் ஒன்றாக எடுத்து, அவற்றில் குறிப்பிடப்பட்டுள்ள வேலைகளைச் செய்து முடிக்கும். பின் திட்ட நிரலிலுள்ள கட்டளைகளின்படி முடிவுகளை நினைவகத்தில் சேமிக்கவோ அல்லது வெளியீட்டுக்கு அனுப்பவோ செய்யும். மேலும் இந்த அலகு நினைவகம் மற்றும் உள்ளீடு/வெளியீடு சாதனங்களுடன் தொடர்பு கொள்ளவோ அல்லது அவற்றைக் கட்டுப்படுத்தவோ செய்யும். மையச் செயலகத்தின் அமைப்பு, படம் (1.2)இல் காட்டப்பட்டுள்ளது. மையச் செயலகத்தின் முக்கியமான பிரிவுகள்

i. கணித மற்றும் வாதியல் அலகு (Arithmetic and logical unit – ALU)
ii. நேர மற்றும் கட்டுப்பாட்டு அலகு (Timing and control unit)
iii. சேமக்கலம் மற்றும் பொது நோக்கப் பதிவேடுகள் (Accumulator and general purpose registers)
iv. சிறப்பு நோக்கப் பதிவேடுகள் (Special purpose registers)

கணித மற்றும் வாதியல் அலகு

எண் கணிதச் செயல்களாகிய கூட்டல், கழித்தல் மற்றும் வாதியல் செயல்களாகிய உம் (AND), அல்ல (OR), தவிர்க்கப்பட்ட அல்ல (Exclusive OR) செயல்களைச் செயல்படுத்துவதே இந்த அலகின் வேலையாகும்.

நேர மற்றும் கட்டுப்பாட்டு அலகு

கணினியின் எல்லா இயக்கங்களையும் இந்த அலகு கட்டுப்படுத்துகிறது. மேலும் மையச் செயலகத்துடன் இணைக்கப்பட்டுள்ள எல்லா அலகுகளையும் கட்டுப்படுத்துகிறது. உள்ளீடு மற்றும் வெளியீடுகளுக்குத் தேவையான நேர சைகைகளை (Timing signals) உருவாக்குகிறது. ஆகவே, இந்த அலகு கணினியின் மூளையாகச் செயல்படுகிறது எனலாம்.

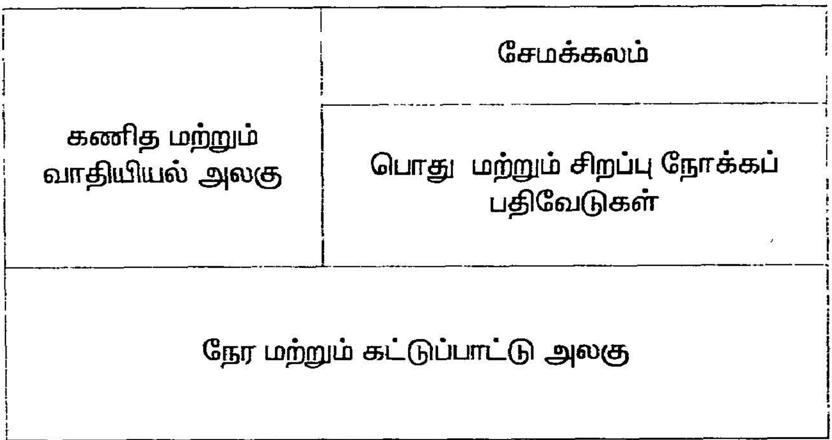
சேமக்கலம் மற்றும் பொது நோக்கப் பதிவேடுகள்

சேமக்கலம் ஒரு பதிவேடாகும். இதில் திட்ட நிரலைச் செயல்படுத்துவதற்குத் தேவையான இயமிகளில் (Operand) ஒரு இயமி வைக்கப்பட்டிருக்கும். மேலும், எண்கணித மற்றும் வாதியல் செயல்களின் முடிவுகள், இதில் சேமித்து வைக்கப்படும்.

கணினி, திட்ட நிரலைச் செயல்படுத்தும்போது வரக்கூடிய இடைநிலை முடிவுகளைத் தற்காலிகமாகப் பொது நோக்கப் பதிவேடுகளில் சேமித்து வைக்கும்.

சிறப்பு நோக்கப் பதிவேடுகள்

திட்டநிரல் எண்ணி (Program counter), அடுக்குச் சுட்டிக்காட்டி (Stack pointer), கட்டளைப் பதிவேடு (Instruction register) மற்றும் நிலைப்பாடு பதிவேடு (Status register) ஆகியவை சிறப்பு நோக்கப் பதிவேடுகளாகும். இவை நுண்முறைப்படுத்தியால் பயன்படுத்தப்படும். திட்ட நிரலரால் மாற்றம் செய்ய முடியாது.



படம் (1.2) மையச் செயலகத்தின் அமைப்பு

1.2.2 நினைவகம் (Memory)

திட்டநிரல்கள், தரவுகள் மற்றும் முடிவுகளைச் சேமித்து வைப்பது நினைவகங்களின் முக்கியமான வேலையாகும். நினைவகம் இரண்டு வகைப்படும்

i. குறைக்கடத்தி நினைவகம் (Semi conductor memory)
ii. காந்த நினைவகம் (Magnetic memory)

குறைக்கடத்தி நினைவகம்

குறைக்கடத்தி நினைவகங்கள், ஒரு கணினியின் முதன்மை நினைவகமாகப் பயன்படுத்தப்படுகின்றன. இந்த நினைவகங்கள் விரைவாகச் செயல்படக்கூடியவை. இவை வேலை செய்வதற்கு மிகக் குறைந்த மின்னாற்றல் பயன்படுத்தப்படுகிறது. இவை இரண்டு வகைப்படும்.

1. நோக்கமின்றி அணுகும் நினைவகம் (Random Access Memory or RAM)
2. படித்தல் மட்டும் நினைவகம் (Read Only Memory or ROM)

நோக்கமின்றி அணுகும் நினைவகம் (RAM)

ஒரு கணினியின் படித்தல் மற்றும் எழுதுதல் நினைவகத்தை RAM எனலாம். சாதாரண இயக்கத்தின்போது செய்திகளை நினைவகத்தில் எழுதவும் மற்றும் படிக்கவும் செய்யலாம். இதற்கு நோக்கமின்றி அணுகும் பண்பு உண்டு. அதாவது நினைவகத்தில் உள்ள எந்த ஒரு இருப்பிடத்தையும் நோக்கமின்றி அணுக முடியும். ஒவ்வொரு நினைவக இருப்பிடத்தையும் அணுகுவதற்கான நேரம், ஒரே மாதிரியாக இருக்கும். இவற்றில் தரவுகளும், திட்ட நிரல்களும் எழுதி வைக்கப்படுகின்றன. இந்த நினைவகம் இரண்டு வகைப்படும். நிலையான நோக்கமின்றி அணுகும் நினைவகம் (Static RAM) மற்றும் சக்தி வாய்ந்த நோக்கமின்றி அணுகும் நினைவகம் (Dynamic RAM).

நிலையான RAMஇல் மின்சாரம் வழங்கும்வரை தகவல்கள் சேமித்து வைக்கப்படும். இந்த நினைவகத்தை அடிக்கடி வலிமையூட்டத் தேவையில்லை. ஆனால் இவை அதிக மின்னாற்றல் கொண்டு வேலை செய்யக் கூடியவை.

சக்திவாய்ந்த RAM மின்சாரம் வழங்கும்போதே சேமித்து வைத்த தகவல்களைச் சிறிது நேரத்தில் இழந்து விடும். அதனால் இந்தச் சக்தி வாய்ந்த RAMஐ, ஒவ்வொரு இரண்டு மில்லி வினாடிகளுக்கும் வலிமையூட்ட வேண்டும். இவை குறைந்த மின்னாற்றல் கொண்டு வேலை செய்யக் கூடியவை. அதோடு இதில் அதிகமான தகவல்களையும் சேமிக்கலாம்.

படித்தல் மட்டும் நினைவகம் (ROM)

ஒரு கணினியில், படித்தல் மட்டும் நினைவகம் (ROM), நிரந்தர சேமிப்பிற்காகப் பயன்படுகிறது. இதற்கும் நோக்கமின்றி அணுகும் பண்பு உண்டு. இதனுள் தகவல்கள், சில்லினைத் தயாரிக்கும் போதே நிரந்தரமாகச் சேமித்து வைக்கப்படும். இதன் உள்ளடக்கத்தைத் தயாரிப்பாளர் முடிவு செய்வார். திட்ட நிரலரால் இந்த நினைவகத்தினுள் எதையும் எழுத முடியாது. மின்சாரம் இல்லாத போதும், இதில் உள்ள தகவல்கள் அழியாது.

திட்ட நிரலாக்கு படித்தல் மட்டும் நினைவகம் (Programmable Read Only Memory - PROM)

இதுவும் அழியா வகை நினைவகம். இந்தச் சில்லினை உருவாக்கும்போது தகவல்கள் எழுதப்படுவதில்லை. இதன் உள்ளடக்கம் பயனீட்டாளரால் நிர்ணயம் செய்யப்படுகிறது. பயனீட்டாளர் இந்த நினைவகத்தில் நிரந்தரத் திட்ட நிரலை எழுதிக் கொள்ளலாம்.

அழியும் திட்ட நிரலாக்கு படித்தல் மட்டும் நினைவகம் (Erasable Programmable Read Only Memory - EPROM)

நுண்முறைப்படுத்திச் சுற்றிலிருந்து இந்த நினைவகத்தை முழுவதுமாக அகற்றி, அதிகச் செறிவுள்ள புறஊதா ஒளியில் பத்திலிருந்து இருபது நிமிடங்களுக்குக் காண்பித்தால், இதன் உள்ளடக்கத்தை அழித்துவிடலாம். இந்த முறையில் நினைவகத்தில் உள்ள ஒரு சிறிய இடத்தை மட்டும் அழிக்க முடியாது. முழுவதுமாகத் தான் அழிக்க முடியும்.

மின்சாரத்தால் அழியும் திட்ட நிரலாக்கு படித்தல் மட்டும் நினைவகம் (Electrically Erasable PROM – EEPROM)

இந்த நினைவகத்தின் உள்ளடக்கத்தை அழிப்பதற்கு நுண்முறைப்படுத்திச் சுற்றிலிருந்து முழுவதுமாக அகற்றிவிட வேண்டும். இதன் உள்ளடகத்தை 10மில்லி வினாடிக்குள் அழித்து விடலாம். தேவைப்பட்டால் ஒரு சிறிய 'பிட்'ஐ மட்டும் தனியாக அழிக்கலாம்.

காந்த நினைவகம்

மின்சாரத்தை நிறுத்தினாலும் இதன் உள்ளடக்கம் அழியாது. குறைக்கடத்தி நினைவகத்தைவிட இதன் வேகம் குறைவானது. இது கணினியில் அதிகமான தகவல்களைச் சேமிக்கப் பயன்படுகிறது.

காந்த நாடா (Magnetic tape), நெகிழ் வட்டு (Floppy disk) மற்றும் வன் வட்டு (Hard disk) ஆகியவை காந்த நினைவகத்தின் முக்கியமான வகைகளாகும்.

1.2.3 உள்ளீட்டகம் (Input device)

தரவையும், திட்ட நிரலையும் பயனீட்டாளரிடமிருந்து பெற்று அவற்றைக் கணினியின் உள்ளே அனுப்பும் சாதனம் இது. மேலும், இந்தச் சாதனம் தரவுகள், கட்டளைகள் மற்றும் சைகைகளை எண்ணிலக்கக் கணினிக்கு ஏற்றவாறு ஈரடி அமைப்பிற்கு மாற்றம் செய்கிறது. விசைப்பலகை (Keyboard), வருடி (Scanner), காந்தப் படிப்பான் (Magnetic Reader) போன்றவை சில உள்ளீட்டகங்களாகும்.

1.2.4 வெளியீட்டகம் (Output device)

கணினி வெளியிடும் தகவல்களைப் பயனீட்டாளர்கள் பயன்படுத்தும் வகையில் மாற்றிக் கொடுப்பவை வெளியீட்டகங்கள் எனப்படும். அச்சுப்பொறி (Printer), திரையகம் (Monitor), எண்ணிலக்க – ஒப்புமை மாற்றி (D/A Converter) போன்றவை சில வெளியீட்டகங்களாகும்.

1.3 ஊர்திகள் (Buses)

உள்ளீடு/வெளியீடு சாதனங்கள் மற்றும் நினைவகம் ஆகியவை மையச் செயலகத்துடன் தகவல் தொடர்பு கம்பிகளின் மூலமாக இணைக்கப்படுகிறது. இந்தத் தகவல் தொடர்பு கம்பிகள் 'ஊர்திகள்' எனப்படும். இவை மூன்று வகைப்படும். i. முகவரி ஊர்தி (Address bus) ii. தரவு ஊர்தி (Data Bus) iii. கட்டுப்பாட்டு ஊர்தி (Control bus). ஊர்திகளின் கட்டமைப்பு, படம் **(1.3)**இல் காட்டப்பட்டுள்ளது.

முகவரி ஊர்தி

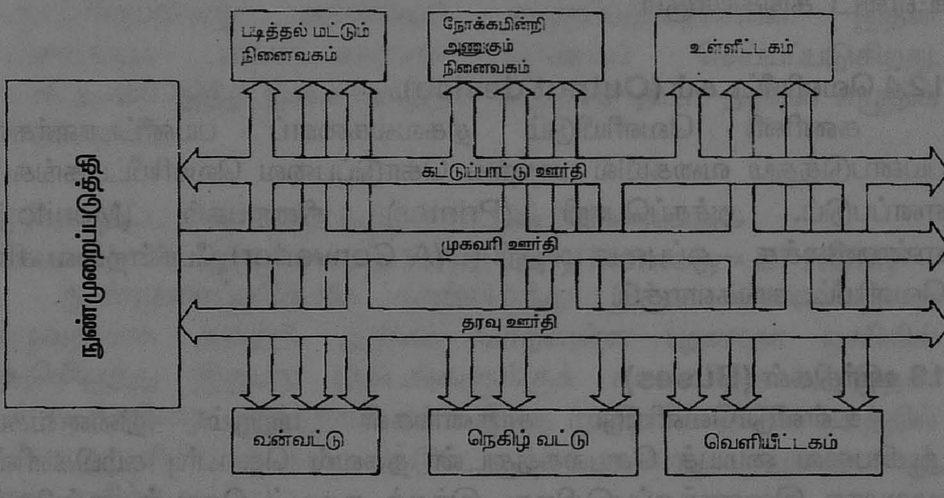
இந்த ஊர்தி மையச் செயலகம், அணுக விரும்பும் நினைவக இருப்பிடத்தின் முகவரியையோ அல்லது உள்ளீடு வெளியீடு சாதனங்களின் முகவரியையோ எடுத்துச் செல்லும். இது ஒரு திசையில் மட்டும் முகவரியை எடுத்துச் செல்லும். இந்த ஊர்தி 16–பிட் அகலம் கொண்டது.

தரவு ஊர்தி

இது இரு திசைகளில் தரவுகளை எடுத்துச் செல்லும். மையச் செயலகம் மற்றும் நினைவகம் அல்லது உள்ளீடு/வெளியீடு சாதனங்களுக்கிடையே தரவுகளை எடுத்துச் செல்ல, இந்த ஊர்தி பயன்படுத்தப்படுகிறது. இதன் அகலம் 8 – பிட் ஆகும்.

கட்டுப்பாட்டு ஊர்தி

இந்த ஊர்தி மையச் செயலகம் மற்றும் அதனுடன் இணைக்கப்பட்டுள்ள மற்ற அலகுகளுக்கிடையே, கட்டுப்பாட்டுச் சைகைகளை எடுத்துச் செல்லும். இது இரு திசைகளில் சைகைகளை எடுத்துச் செல்லும். இதன் அகலம் 8 – பிட் ஆகும்.

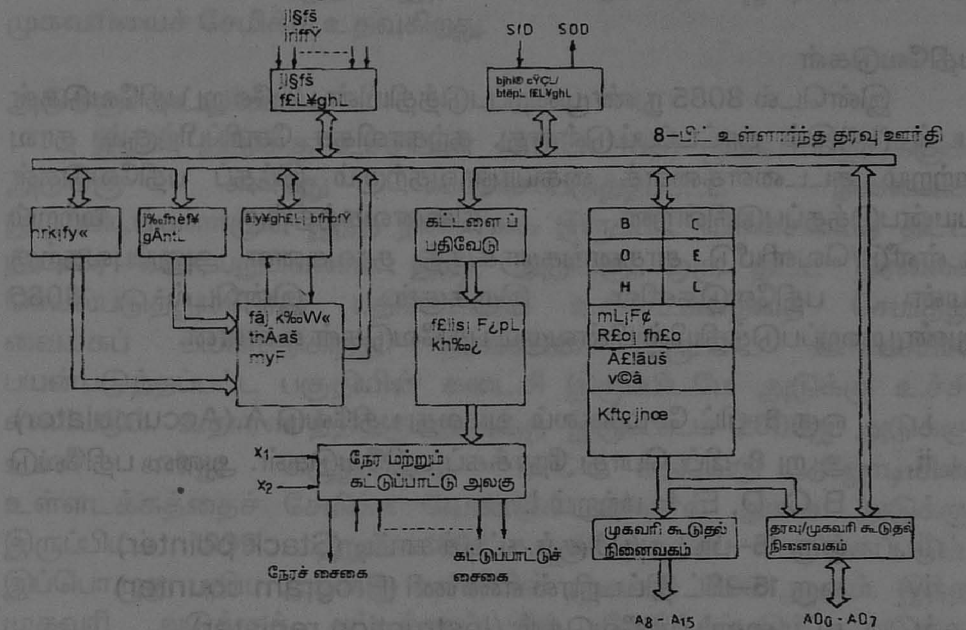


படம் (1.3) ஊர்திகளின் கட்டமைப்பு

1.4. நுண்முறைப்படுத்தியின் கட்டுமான அமைப்பு

நுண்முறைப்படுத்தி, ஒரு நுண்கணினியின் மையச் செயலகமாகும். இது நுண்கணினியின் இருதயமாகக் கருதப்படுகிறது. இன்டெல் 8085 ஒரு 8–பிட் நுண்முறைப்படுத்தியாகும். இது பெரிய அளவிலான தனித் தொகுப்புச் சில்லில் தயாரிக்கப்பட்டதாகும். இந்தத் தொகுப்புச் சில்லில் 40 மின்முனைகள் உள்ளன. இது வேலை செய்வதற்கு +5V நேர்மின்னழுத்தம் தேவைப்படுகிறது. இதனின் கடிகார வேகம் 3MHz ஆகும். இந்த நுண் முறைப்படுத்திக்கு 80 கட்டளைகளும் மற்றும் 246 இயக்கக் குறியீடுகளும் உள்ளன. இதன் கட்டமைப்பு, படம் (1.4)இல் காட்டப்பட்டுள்ளது. இது மூன்று பிரிவுகளைக் கொண்டது. i. கணித மற்றும் வாதியல் அலகு ii. நேர மற்றும்

கட்டுப்பாட்டு அலகு iii. பல வகையான பதிவேடுகள். இதன் விளக்கத்தைப் பின்வரும் பிரிவுகளில் பார்ப்போம்.



படம் (1.4) இன்டெல் 8085இன் கட்டமைப்பு

கணித மற்றும் வாதியல் அலகு

இந்த அலகு, பின்வரும் எண்கணித மற்றும் வாதியல் இயக்கங்களைச் செயல்படுத்தும்.

i. கூட்டல் ii. கழித்தல் iii. வாதியல் உம் iv. வாதியல் அல்ல v. வாதியல் தவிர்க்கப்பட்ட அல்ல vi. ஒரு பிட் குறைத்தல் vii. ஒரு பிட் அதிகரித்தல் viii. தரவை இடதுபக்கம் நகர்த்துதல்.

நேர மற்றும் கட்டுப்பாட்டு அலகு

இது மையச் செயலகத்தின் ஒரு பகுதியாகும். கட்டளைகளைச் செயல்படுத்தத் தேவையான நேர மற்றும் கட்டுப்பாட்டு சைகைகளை உற்பத்தி செய்கிறது. மையச் செயலகத்திற்கும் மற்றும் புற அலகுகளுக்கிடையேயும் தரவு ஓட்டத்தைக் கட்டுப்படுத்துகிறது. நினைவகம் மற்றும் உள்ளீடு/வெளியீடு சாதனங்களின் இயக்கங்களுக்குத் தேவையான நிலைப்பாடு, நேர மற்றும் கட்டுப்பாட்டுச் சைகைகளைக் கொடுக்கிறது. மையச் செயலகம் மற்றும் அதனுடன் இணைக்கப்பட்டுள்ள அலகுகளின் எல்லா

இயக்கங்களையும் கட்டுப்படுத்துகிறது. இந்த அலகு கணினியின் மூளையாகச் செயல்படுகிறது.

பதிவேடுகள்

இன்டெல் 8085 நுண்முறைப்படுத்தியின் பல்வேறு பதிவேடுகள் படம் (1.4)இல் காட்டப்பட்டுள்ளது. தற்காலிகச் சேமிப்பிற்கும், தரவு மற்றும் கட்டளைகளைக் கையாளுவதற்கும் இந்தப் பதிவேடுகள் பயன்படுத்தப்படுகின்றன. நினைவகத்திற்கு மற்றும் உள்ளீடு/வெளியீடு சாதனங்களுக்குத் தரவுகளை அனுப்புவதற்கு முன் பதிவேடுகளில் இருக்கும். இன்டெல் 8085 நுண்முறைப்படுத்தியில் பின்வரும் பதிவேடுகள் உள்ளன.

i. ஒரு 8–பிட் சேமக்கலம் அல்லது பதிவேடு A.(Accumulator)
ii. ஆறு 8–பிட் பொது நோக்கப் பதிவேடுகள். அவை பதிவேடு B,C, D, E, H மற்றும் L.
iii. ஒரு 16–பிட் அடுக்குச் சுட்டிக்காட்டி (Stack pointer)
iv. ஒரு 16–பிட் திட்டநிரல் எண்ணி (Program counter)
v. கட்டளைப் பதிவேடுகள் (Instruction register)
vi. நிலைப்பாடு பதிவேடுகள் (Status register)
vii. தற்காலிகப் பதிவேடுகள் (Temporary register)

சேமக்கலம்

பதிவேடு A இன்டெல் 8085வின் சேமக்கலமாகும். திட்ட நிரலைச் செயல்படுத்தும்போது தற்காலிகச் சேமிப்பிற்காக இது பயன்படுத்தப்படுகிறது. இயமிகளில் ஒரு இயமி இதில் சேமித்து வைக்கப்படும். மற்றொரு இயமி பிற பதிவேடுகளிலோ அல்லது நினைவகத்திலோ வைக்கப்படும். எண்கணித மற்றும் வாதியல் செயல்களின் முடிவுகள் சேமக்கலத்தில் இருக்கும்.

பொது நோக்கப் பதிவேடுகள்

இன்டெல் 8085 நுண்முறைப்படுத்தியில் ஆறு 8–பிட் பொது நோக்கப் பதிவேடுகள் உள்ளன. அவை பதிவேடு B, C, D, E, H மற்றும் L ஆகு ம். இவற்றின் உள்ளடக்கம் திட்ட நிரலரால் வரையறுக்கப்படுகிறது. 16–பிட் தரவுகளைச் சேமிக்க வேண்டுமென்றால், இரண்டு 8–பிட் பதிவேடுகளை ஒன்றாக இணைத்து கொள்ளலாம். ஒன்றாக இணைக்கப்பட்ட இரண்டு 8–பிட் பதிவேடுகளைப் பதிவேடு ஜோடி எனலாம். இன்டெல் 8085

நுண்முறைப்படுத்தியின் அனுமதிக்கப்பட்ட பதிவேடு ஜோடிகள் B-C, D-E மற்றும் H-L ஆகும். இதில் H-L பதிவேடு ஜோடி, நினைவக முகவரியைச் சேமிக்க உதவுகிறது.

அடுக்குச் சுட்டிக்காட்டி

இதில் அடுக்கு உச்சியின் (Stack top) முகவரி இருக்கும். அடுக்கு என்பது வரிசைப்படுத்தப்பட்ட நினைவக இருப்பிடங்களாகும். இந்த நினைவக இருப்பிட வரிசையைத் திட்ட நிரலரே வரையறுக்கலாம். இந்த அடுக்கு, ஒரு திட்ட நிரலைச் செயல்படுத்தும்போது பதிவேட்டின் உள்ளடக்கத்தை சேமித்து வைக்கப் பயன்படுகிறது. நினைவக இருப்பிட வரிசையில் பயன்படுத்தப்பட்ட பகுதியின் கடைசி இருப்பிடமே அடுக்கு உச்சி எனப்படும். உதாரணத்திற்கு அடுக்கு இருப்பிடம் 2000ஐ அடுக்கு உச்சி என்று கருதுவோம். B-C பதிவேடு ஜோடியின் உள்ளடக்கத்தைச் சேமிக்க வேண்டும் என்றால், அவை அடுக்கு இருப்பிடம் 1999 மற்றும் 1998 ஆகியவற்றில் சேமிக்கப்படும். இப்பொழுது புதிய அடுக்கு உச்சியின் முகவரி 1998 ஆகும். இந்த முகவரி அடுக்குச் சுட்டிக்காட்டியில் சேமிக்கப்படும். அடுத்து வரக்கூடிய தரவுகளை அடுக்கு இருப்பிடம் 1997லிருந்து சேமிக்கப்படும்.

திட்ட நிரல் எண்ணி

திட்ட நிரலில், அடுத்த கட்டளையின் முகவரி இந்த எண்ணியில் இருக்கும். மையச் செயலகம் நினைவகத்திலிருந்து ஒரு கட்டளையை எடுத்து, அதை செயல்படுத்திப் பிறகு திட்ட நிரல் எண்ணியின் உள்ளடகத்தை அதிகரிக்கச் செய்யும். அடுத்த கட்டளைச் சுற்றில், மையச் செயலகம் திட்ட நிரல் எண்ணியில் குறிப்பிட்டுள்ளபடி அடுத்த கட்டளையை நினைவகத்திலிருந்து எடுத்துச் செயல்படுத்தும். திட்ட நிரல் எண்ணியின் உள்ளடக்கம் மாறாமல் இருக்கும்வரை, கட்டளைகள் ஒன்றன்பின் ஒன்றாகச் செயல்படுத்தப்படும்.

கட்டளைப் பதிவேடு

கட்டளைகள் ஈரடி அமைப்பிற்கு மாறும்வரை கட்டளை பதிவேட்டில் இருக்கும்.

நிலைப்பாடு பதிவேடு

ஐந்து நிலைமாறிகளின் தொகுப்பு, நிலைப்பாடு கொடிகளாகச் செயல்படுகிறது. ஒவ்வொரு நிலைமாறியிலும் ஒரு பிட் இருக்கும். இவை எண்கணித மற்றும் வாதியல் இயக்கங்களில் ஏற்படும் நிபந்தனைகளைக் குறிப்பதாக இருக்கும். இன்டெல் 8085 நுண்முறைப்படுத்தியில் உள்ள நிலைப்பாடு கொடிகள் i. சுமை ii. சுழி iii. குறியீடு iv. சமநிலை மற்றும் v. துணைச் சுமை.

சுமை (carry)

கூட்டல் இயக்கத்தின்போது சுமை ஏற்பட்டாலோ அல்லது கழித்தல் இயக்கத்தின்போது கடன் பெறப்பட்டாலோ, இந்தக் கொடி ஒன்று என்று அமைவு பெறும்.

சுழி (Zero)

எண்கணித அல்லது வாதியல் முடிவுகள் சுழியாக இருந்தால், இந்தக் கொடி ஒன்று என்று அமைவு பெறும்.

குறியீடு (Sign)

எண்கணித அல்லது வாதியல் முடிவில் பெரும முக்கியத்துவம் வாய்ந்த பிட் ஒன்றாக இருந்தால், இந்தக் கொடி ஒன்று என்று அமைவு பெறும்.

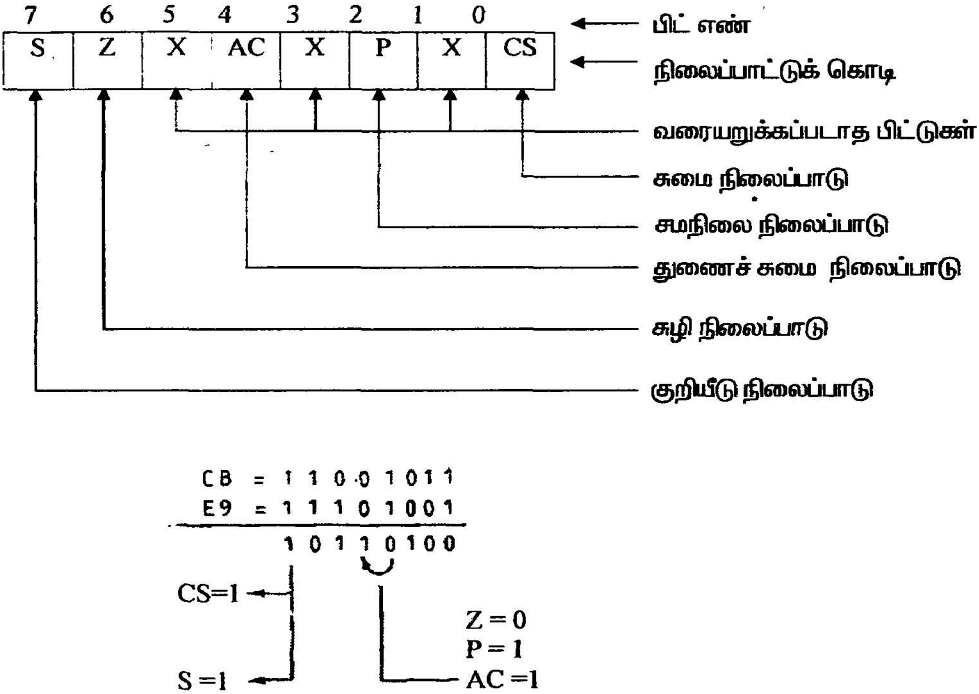
சமநிலை (Parity)

எண்கணித அல்லது வாதியல் முடிவுகளில் இரட்டைப் படை ஒன்றுகள் இருந்தால், இந்தக் கொடி ஒன்று என்று அமைவு பெறும்.

துணைச் சுமை (Auxillary carry)

எண்கணித இயக்கத்தில் மூன்றாவது பிட்லிருந்து நான்காவது பிட்க்குச் சுமை இருந்தால், இந்தக் கொடி ஒன்று என்று அமைவு பெறும்.

இன்டெல் 8085 நுண்முறைப்படுத்தியில் நிலைப்பாடு கொடிகளின் அமைப்பு, படம் (1.5)இல் காட்டப்பட்டுள்ளது.



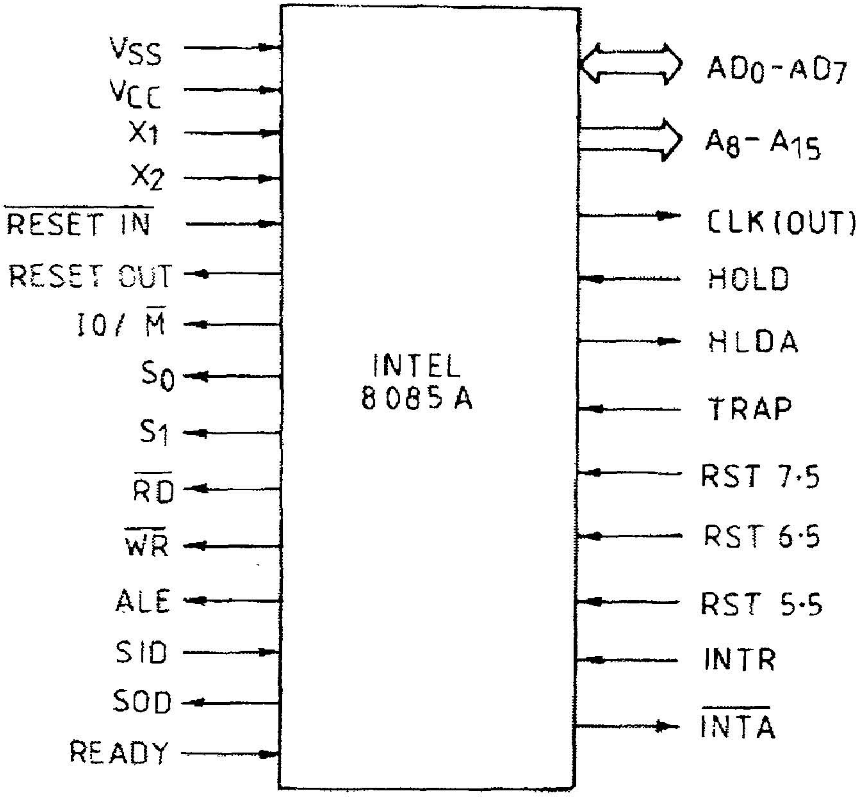
படம் (1.5) நிலைப்பாடு கொடிகளின் அமைப்பு

திட்ட நிரல் நிலைப்பாடு வார்த்தை (Program status word)

படம் (1.5)இல் ஐந்து பிட்டுகள், ஐந்து நிலைப்பாடு கொடியைக் காட்டுகின்றன. மீதி மூன்று பிட்டுகள் வரையறுக்கப்படாத பிட்டுகள். மேலே கூறப்பட்ட 8–பிட்டுகளின் தொகுப்பு, திட்ட நிரல் நிலைப்பாடு வார்த்தை எனப்படும்.

1.5 இன்டெல் 8085 நுண்முறைப்படுத்தியின் முனை அமைப்பு

நுண்முறைப்படுத்தியின் முனை அமைப்பு, படம் (1.6)இல் காட்டப்பட்டுள்ளது. ஒவ்வொரு முனையிலிருந்தும் வெளியேறுகிற மற்றும் முனையின் வழியாக மையச் செயலகத்தினுள் செல்லுகிற சைகைகளைப் பற்றிய விளக்கத்தைக் காண்போம்.



படம் (1.6) நுண்முறைப்படுத்தியின் முனை அமைப்பு

$A_8 - A_{15}$

இந்த முனைகள் முகவரி ஊர்தியாகச் செயல்படுகின்றன. இவை நினைவக முகவரியில் உள்ள பெரும முக்கியத்துவம் வாய்ந்த பிட்டுகள் (Most Significant Bits) அல்லது உள்ளீடு மற்றும் வெளியீடு முகவரியை எடுத்துச் செல்லும்.

$AD_0 - AD_7$

இந்த முனைகள் முகவரி / தரவு ஊர்தியாகச் செயல்படுகின்றன. இவை இரண்டு தேவைகளுக்காகப் பயன்படுத்தப்படுகிறது. கருவிச் சுற்றின் (Machine cycle) முதல் கடிகாரச் சுற்றில் (Clock cycle), நினைவக முகவரியில் உள்ள சிறும முக்கியத்துவம் வாய்ந்த பிட்டுகளை (Least Significant Bits) எடுத்துச் செல்லும். இரண்டு மற்றும் மூன்றாம் கடிகாரச் சுற்றில் தரவுகளை எடுத்துச் செல்லும். இந்த நுணுக்கம், பலவின் ஒன்றாக்குதல் (Multiplexing) எனப்படுகிறது.

ALE (Address Latch Enable)

இது முகவரி தாழ் செயலியக்கச் சைகையாகும். கருவிச் சுற்றின் முதல் கடிகாரச் சுற்றின்போது இந்தச் சைகை உயர்நிலைக்குச் செல்லும். இதனால் நினைவக முகவரியில் உள்ள சிறும முக்கியத்துவம் வாய்ந்த பிட்டுகள் நினைவகத்தில் வைக்கப்படும்.

IO/$\overline{M}$

இது ஒரு நிலைப்பாடு சைகையாகும். முகவரி ஊர்தியில் உள்ள முகவரி நினைவகத்திற்கா அல்லது உள்ளீடு வெளியீடு சாதனங்களுக்கா என்று வித்தியாசப்படுத்திக் காட்டக்கூடியது. இந்தச் சைகை உயர்நிலைக்குச் சென்றால் முகவரி ஊர்தியில் உள்ள முகவரி உள்ளீடு/வெளியீடு சாதனங்களுக்கு என்றும், இதே சைகை தாழ்நிலைக்குச் சென்றால் முகவரி நினைவகத்திற்கு என்றும் முடிவு செய்யப்படும்.

S_0S_1

இதுவும் நிலைப்பாடு சைகைகளாகும். இவை வெவ்வேறு செயற்பாடுகளைப் பிரித்துப் பார்க்கக்கூடியது.

S_1	S_0	செயற்பாடுகள்
0	0	செயற்பாட்டை நிறுத்துதல் (HALT)
0	1	நினைவகத்தில் எழுதுதல் (WRITE)
1	0	நினைவகத்தைப் படித்தல் (READ)
1	1	நினைவகத்திலிருந்து கட்டளையைக் கொணர்தல் (FETCH)

$\overline{RD}$

படிக்கும் செயலை இந்தச் சைகை கட்டுப்படுத்தும். இந்தச் சைகை தாழ்நிலைக்குச் செல்லும்போது தேர்ந்தெடுக்கப்பட்ட நினைவகமோ அல்லது உள்ளீடு/வெளியீடு சாதனங்களோ படிக்கப்படும்.

$\overline{WR}$

எழுதும் செயலை இந்தச் சைகை கட்டுப்படுத்தும். இந்தச் சைகை தாழ்நிலைக்குச் செல்லும்போது தேர்ந்தெடுக்கப்பட்ட நினைவகத்தில் அல்லது உள்ளீடு/வெளியீடு சாதனங்களில், தரவு ஊர்தியில் உள்ள தரவுகள் எழுதப்படும்.

READY

இந்தச் சைகை நுண்முறைப்படுத்தியால் பயன்படுத்தப்படுகிறது. இந்தச் சைகை மூலம் வெளிப்புற அலகுகள் தரவுகளைத் தரத் தயாராக இருக்கின்றனவா என்பதைத் தெரிந்துகொள்ளும். க்ஷவளிப்புற அலகுகள் தரவுகளைத் தரத் தயாராக இருந்தால், இந்தச் சைகை உயர்நிலையில் இருக்கும்.

HOLD

இந்தச் சைகை, நுண்முறைப்படுத்தியோடு இணைக்கப்பட்ட பிற அலகுகளுக்கு முகவரி மற்றும் தரவு ஊர்திகள் தேவைப்படுகின்றன என்பதை உணர்த்துகிறது. இந்தச் சைகையைப் பெற்றவுடன் ஊர்திகளைப் பயன்படுத்துவதை நுண்முறைப்படுத்தி விட்டுவிடுகிறது. ஆனால் உள்ளார்ந்த செயல்கள் தொடரும். இந்தச் சைகை விலகியவுடன், நுண்முறைப்படுத்தி, திரும்பவும் ஊர்திகளைப் பயன்படுத்தத் தொடங்கிவிடும்.

HLDA

·HOLD சைகையைப் பெற்றுக் கொண்டதற்கு அடையாளமாக, நுண்முறைப்படுத்தி திரும்ப அனுப்பும் சைகையே HLDA ஆகும்.

INTR

இது ஒரு தடங்கல் சைகையாகும். இந்தச் சைகை உயர்நிலைக்குச் செல்லும்போது திட்ட நிரல் எண்ணியின் உள்ளடக்கம் அதிகரிக்கப்பட மாட்டாது. நுண்முறைப்படுத்தி வழக்கமான தொடர்வரிசையில் உள்ள கட்டளைகளைச் செயல்படுத்துவதைத் தற்காலிகமாக நிறுத்திவிடும். பிறகு CALL கட்டளையைச் செயல்படுத்தத் தயாராகிவிடும். மென்பொருளைக் கொண்டு இந்தச் சைகையை செயலியக்கவோ அல்லது செயல் முடுக்கவோ செய்யலாம். தரவுகளை நுண்முறைப்படுத்திக்கு மாற்றம் செய்வதற்கு, உள்ளீடு/வெளியீடு அலகுகள் இந்தச் சைகையைப் பயன்படுத்திக் கொள்கின்றன.

$\overline{\text{INTA}}$

தடங்கல் சைகையைப் பெற்றுக் கொண்டதற்கு அடையாளமாக நுண்முறைப்படுத்தி திரும்ப அனுப்பும் சைகை $\overline{\text{INTA}}$ ஆகும்.

RST5.5, 6.5, 7.5 மற்றும் TRAP

இவை தடங்கல் சைகைகளாகும். இவற்றில் ஏதாவது ஒன்று நுண்முறைப்படுத்திக்கு வந்தால், உடனே நுண்முறைப்படுத்தியானது, அடுத்த கட்டளையைக் கீழே கொடுக்கப்பட்ட நிரந்தர நினைவக இருப்பிடங்களிலிருந்து எடுத்துச் செயல்படுத்தும்.

தடங்கல் சைகை	நினைவக இருப்பிடம்
TRAP	0024
RST 5.5	002C
RST 6.5	0034
RST 7.5	003C

இவற்றில் TRAP என்ற தடங்கல் சைகைக்கு அதிக முக்கியத்துவம் கொடுக்கப்படுகிறது. இந்தச் சைகையை மறைக்க முடியாது. கீழே கொடுக்கப்பட்ட வரிசைப்படி தடங்கல் சைகைகளுக்கு முக்கியத்துவம் கொடுக்கப்படுகிறது.

TRAP (அதிக முக்கியத்துவம் உடையது)
RST 7.5
RST 6.5
RST 5.5
INTR (குறைந்த முக்கியத்துவம் உடையது)

$\overline{\text{RESET IN}}$

இந்தச் சைகை திட்ட நிரல் எண்ணியின் உள்ளடக்கத்தைச் சுழியாக்குகிறது. அதாவது மறு அமைவு செய்கிறது. இது கட்டளைப் பதிவேட்டைத் தவிர, மற்ற பதிவேடுகளைப் பாதிப்பதில்லை. இந்தச் சைகை தாழ்நிலையில் இருக்கும்வரை மையச் செயலகமும் மறு அமைவு நிலையில் இருக்கும்.

$\overline{\text{RESET OUT}}$

இந்தச் சைகை புறவியலான அலகுகளுக்கு, மையச் செயலகம் மறுஅமைவு நிலையில் இருக்கிறது என்பதை உணர்த்தும்.

X_1X_2

இந்த முனைகள் படிக அலையியற்றியுடன் இணைக்கப்படுகின்றன. படிக அலையியற்றி, நுண்முறைப்படுத்தி இயங்குவதற்குத் தேவையான கடிகாரத் துடிப்புகளை உற்பத்தி செய்கிறது.

CLK

இது பயனாளிக்கான கடிகார வெளியீடாகும். இதைப் பிற எண்ணிலக்கத் தொகுப்புச் சுற்றுகளுக்குப் பயன்படுத்தலாம். இதன் அதிர்வெண்ணும், நுண்முறைப்படுத்தி இயக்கத்தின் அதிர்வெண்ணும் ஒரே மாதிரி இருக்கும்.

SID

இது தொடர் உள்ளீட்டிற்கான தரவு கோடாகும். RIM கட்டளையைச் செயல்படுத்தும்போது, இந்தக் கோட்டிலுள்ள தரவு சேமக்கலத்தில் உள்ள ஏழாவது பிட்டில் வைக்கப்படும்.

SOD

இது தொடர் வெளியீட்டிற்கான தரவு கோடாகும். SIM கட்டளையைச் செயல்படுத்தும்போது, சேமக்கலத்தில் ஏழாவது பிட்டிலுள்ள தரவு SOD கோட்டில் வைக்கப்படும்.

VCC

இந்த முனை +5V நேர்மின்னழுத்தத்துடன் இணைக்கப்படுகிறது.

VSS

இந்த முனை தரை இணைப்புச் செய்யப்படுகிறது.

1.6 இன்டெல் 8085 கட்டளைகள்

கணினியானது கொடுக்கப்பட்ட தரவுகளைக் கொண்டு ஒரு குறிப்பிட்ட செயலைக் கட்டளைக்குத் தகுந்தவாறு செய்து முடிக்கும். ஒவ்வொரு கட்டளைக்கும் இரண்டு பகுதி உண்டு. ஒன்று இயக்கக் குறியீடு (Opcode). மற்றொன்று இயமி (Oporand). கட்டளையின் முதல்பகுதி, கணினி செய்து முடிக்கக்கூடிய வேலையைக் குறிப்பிடுகிறது. இது இயக்கக் குறியீடு எனப்படுகிறது. கட்டளையின் இரண்டாம் பகுதி தரவுகளைக் குறிப்பிடுகிறது. இது இயமி எனப்படுகிறது. கட்டளையில் உள்ள இயமி, 8–பிட் மற்றும் 16–பிட் தரவுகளாகவோ அல்லது 8–பிட் மற்றும் 16–பிட் முகவரியாகவோ இருக்கலாம். சில கட்டளைகளில் இயமி மறைந்து காணப்படும். சில கட்டளைகளில் இயமி ஒரு பதிவேடாக இருக்கும். தரவுகள் பதிவேட்டின் உள்ளடக்கமாக இருக்கும்.

கட்டளையின் அளவு

ஒரு எண்ணிலக்கக் கணினி, ஈரடி குறியீட்டில் (கருவி குறியீடு) எழுதப்பட்ட கட்டளைகளைப் புரிந்துகொள்ளும். எல்லா கட்டளைகளின் கருவி குறியீடுகளும் ஒரே அளவாக இருப்பதில்லை. அவைகளின் அளவைப் பொருத்துக் கட்டளைகளை மூன்று வகையாகப் பிரிக்கலாம்.

i. 1–பைட் கட்டளை

ii. 2 – பைட் கட்டளை

iii. 3 – பைட் கட்டளை

1 – பைட் கட்டளை

(உ.ம்) MOV A, B அதாவது B பதிவேட்டில் உள்ளதை A பதிவேட்டிற்கு நகர்த்த வேண்டும். இந்தக் கட்டளைக்கான இயக்கக் குறியீடு 78H ஆகும். இந்தக் குறியீடு எந்த இயக்கம் நடைபெற வேண்டும் என்பதைச் சொல்வதோடு, தரவுகள் இருக்கும் பதிவேடுகளையும் குறிப்பிடுகிறது. இயக்கக் குறியீடு 78Hன் ஈரடி அமைப்பு 01111000 ஆகும். இதில் முதல் இரண்டு பிட்டுகள் அதாவது 01 நகர்த்துதல் இயக்கத்தைக் குறிக்கிறது. அடுத்த மூன்று பிட்டுகள் (111) பதிவேடு Aஇன் ஈரடி குறியீடாகும். கடைசி மூன்று பிட்டுகள் (000) பதிவேடு Bஇன் ஈரடி குறியீடாகும்.

2–பைட் கட்டளை

இந்தக் கட்டளையில், முதல் பைட் கட்டளையின் இயக்கக் குறியீடாகவும், இரண்டாவது பைட் தரவுகளாகவோ அல்லது முகவரியாகவோ இருக்கும். 2–பைட் கட்டளைகள் அடுத்தடுத்துள்ள இரண்டு நினைவக இருப்பிடங்களில் சேமிக்கப்படும். (உ.ம்) MVI B, 05.அதாவது 05 என்ற எண்ணைப் பதிவேடு Bக்கு நகர்த்த வேண்டும். குறியீடு முறையில் இந்தக் கட்டளையை 06, 05 எனக் குறிப்பிடலாம். இதில் முதல் பைட் 06, MVI B என்ற பகுதியின் இயக்கக் குறியீடாகும். இரண்டாவது பைட் 05 தரவாகும்.

3– பைட் கட்டளை

இதில் முதல் பைட் இயக்கக் குறியீடாக இருக்கும். இரண்டு மற்றும் மூன்றாவது பைட்டுகள் 16–பிட் தரவாகவோ அல்லது 16–பிட் முகவரியாகவோ இருக்கும் (உ.ம்) LXI H, 2400 H. அதாவது H-L பதிவேடு ஜோடியில், நினைவக இருப்பிட முகவரியான 2400H ஐ

உடனடியாகப் பளுவேற்ற வேண்டும். குறியீடு முறையில் இந்த கட்டளையை 21,00,24 என்று குறிப்பிடலாம். இதில் முதல் பைட் 21, LXIHஇன் இயக்கக் குறியீடாகும். இரண்டாவது பைட் 00 முகவரியின் சிறும முக்கியத்துவம் வாய்ந்த பிட்டுகளாகும். இவை L பதிவேட்டில் சேமிக்கப்படும். மூன்றாவது பைட் 24 முகவரியின் பெரும முக்கியத்துவம் வாய்ந்த பிட்டுகளாகும். இவை H பதிவேட்டில் சேமிக்கப்படும்.

1.7 முகவரியிடல் வகைகள் (Addressing Modes)

ஒவ்வொரு கட்டளையையும் செயல்படுத்துவதற்குச் சில தரவுகள் தேவைப்படுகின்றன. கட்டளைகளுக்குத் தேவையான தரவுகளைத் தெளிவாகக் குறிப்பிட பல யுக்திகள் கையாளப்படுகின்றன. இந்த யுக்திகளே முகவரியிடல் வகைகள் எனப்படும். இன்டெல் 8085 நுண்முறைப்படுத்தியில், பின்வரும் முகவரியிடல் வகைகள் பயன்படுத்தப்படுகின்றன.

i. நேரடி முகவரியிடல் (Direct addressing)
ii. பதிவேடு முகவரியிடல் (Register addressing)
iii. பதிவேடு மறைமுக முகவரியிடல் (Register indirect addressing)
iv. உடனடி முகவரியிடல் (Immediate addressing)
v. உள்ளடக்க முகவரியிடல் (Implicit addressing)

நேரடி முகவரியிடல்

இந்த வகையில் இயமியின் முகவரி, கட்டளையிலே கொடுக்கப்பட்டிருக்கும். (உ.ம்) STA 2400H. அதாவது சேமக்கலத்தில் உள்ளதை 2400H என்ற முகவரியையுடைய நினைவக இருப்பிடத்தில் சேமிக்க வேண்டும். இந்தக் கட்டளைக்கான தரவுகள், 2400H என்ற முகவரியையுடைய நினைவக இருப்பிடத்தில் இருக்கும். முகவரி கட்டளையிலே கொடுக்கப்பட்டிருக்கிறது. குறியீடு முறையில் STA 2400H என்ற கட்டளையை 32,00,24 என்று குறிப்பிடலாம். இதில் இரண்டாவது மற்றும் மூன்றாவது பைட்டுகள் நினைவக இருப்பிடத்தின் முகவரியைக் குறிக்கிறது.

பதிவேடு முகவரியிடல்

இந்த வகையில் இயமியானது (தரவு) பொது நோக்கப் பதிவேடுகளில் இருக்கும். இயக்கக் குறியீடு, எந்த இயக்கத்தைச் செயல்படுத்த வேண்டும் என்று கூறுவதோடு பதிவேடுகளின் முகவரியையும் குறிப்பிடுகிறது. (உ.ம்) MOV A,B. இந்தக் கட்டளையின் இயக்கக் குறியீடு 78H ஆகும். இதன் ஈரடி அமைப்பு 01111000 ஆகும். இதில் முதல் இரண்டு பிட்டுகள் (01) MOV இயக்கத்தைக் குறிப்பிடுகிறது. அடுத்த மூன்று பிட்டுகள் (111) A பதிவேட்டின் முகவரியையும், கடைசி மூன்று பிட்டுகள் (000) B பதிவேட்டின் முகவரியையும் குறிக்கிறது.

பதிவேடு மறைமுக முகவரியிடல்

இந்த வகையில் இயமியின் முகவரி பதிவேடு ஜோடிகளால் குறிப்பிடப்படுகிறது. (உ.ம்)

LXI H, 2500 H	:	H-L பதிவேடு ஜோடிகளில், 2500H என்ற, நினைவக இருப்பிட முகவரியை உடனடியாகப் பளுவேற்ற வேண்டும்.
MOV A, M	:	நினைவக இருப்பிடத்தில் உள்ளதைச் சேமகலத்திற்கு நகர்த்த வேண்டும்.
HLT	:	தற்காலிக நிறுத்தம்

மேலே கூறப்பட்ட திட்டநிரலில், MOV A, M என்ற கட்டளை, பதிவேடு மறைமுக முகவரியிடலுக்கான உதாரணமாகும். இந்தக் கட்டளையில் இயமி, நினைவகத்தில் இருக்கும். நினைவகத்தின் முகவரி, கட்டளையில் வெளிப்படையாகக் கொடுக்கப்படவில்லை. ஆனால், முகவரி H-L பதிவேடு ஜோடியில் வைக்கப்பட்டுள்ளது.

உடனடி முகவரியிடல்

இந்த வகையில் கட்டளையிலே இயமி குறிப்பிடப்பட்டிருக்கும். (உ.ம்) MVI A, 05. அதாவது 05 என்ற எண்ணைச் சேமக்கலத்திற்கு உடனடியாக நகர்த்த வேண்டும். குறியீடு முறையில் இந்தக் கட்டளையை 3E, 05 என குறிப்பிடலாம். இதில் இரண்டாவது பைட் (05) தரவாகும்.

உள்ளடக்க முகவரியிடல்

சில கட்டளைகள் சேமக்கலத்தின் உள்ளடக்கத்தைக் கொண்டு இயங்கும். அவ்வகைக் கட்டளைகளுக்கு இயமியின் முகவரி தேவையில்லை. (உ.ம்) CMA, RAL, RAR.

1.8 இன்டெல் 8085 நுண்முறைப்படுத்தியின் கட்டளை வகைகள்

கணினி ஒரு குறிப்பிட்ட செயலைச் செய்யத் தூண்டுகின்ற உள்ளீடு, கட்டளை எனப்படும். ஒரு குறிப்பிட்ட வேலையை முழுமையாகச் செய்து முடிப்பதற்கான கட்டளைத் தொடர், திட்ட நிரல் (Program) எனப்படும். ஒவ்வொரு நுண்முறைப்படுத்தியும் தனிப்பட்ட கட்டளை அமைப்பு கொண்டிருக்கும். கட்டளைகளை நமது வசதிக்காக ஐந்து தொகுப்புக்களாகப் பிரிக்கலாம்.

i. தரவு மாற்றும் தொகுப்பு (Data transfer group)
ii. கணிதத் தொகுப்பு (Arithmetic group)
iii. வாதியல் தொகுப்பு (Logical group)
iv. கிளைத் தொகுப்பு (Branch group)
v. அடுக்கு, உள்ளீடு வெளியீடு மற்றும் கருவி கட்டுப்பாட்டுத் தொகுப்பு (Stack, I/O, Machine control group)

தரவு மாற்றும் தொகுப்பு

MOV r_1, r_2

$[r_1] \leftarrow [r_2]$

பதிவேடு r_2ல் உள்ளதைப் பதிவேடு r_1ற்கு நகர்த்த வேண்டும் (உ.ம்) MOV A,B.

MOV r, M

[r] ←[H-L]

நினைவக இருப்பிடத்தில் உள்ளதைப் பதிவேடு rக்கு நகர்த்த வேண்டும். நினைவக இருப்பிடத்தின் முகவரி, H-L பதிவேடு ஜோடியில் இருக்கும்.

```
(உ.ம்) LXI H, 2000H
       MOV B, M
       HLT
```

MOV M, r

[[H-L]] ← [r]

பதிவேடு r இல் உள்ளதை நினைவக இருப்பிடத்திற்கு நகர்த்த வேண்டும். நினைவக இருப்பிடத்தின் முகவரி, H-L பதிவேடு ஜோடியில் இருக்கும்.

(உ.ம்) MOV M,C.

MVI r, data

[r] ← data

தரவுகளை உடனடியாகப் பதிவேடு rக்கு நகர்த்த வேண்டும். (உ.ம்) MVI r, 05.

MVI M, data

[[H-L]] ← data

தரவுகளை உடனடியாக நினைவகத்திற்கு நகர்த்த வேண்டும். (உ.ம்) MVI M, 08

LXI rp, data 16

[rp] ← data 16 bits

16-பிட் தரவுகளை உடனடியாக H-L பதிவேடு ஜோடியில் பளுவேற்ற வேண்டும். (உ.ம்) LXI H, 2500H. இந்தக் கட்டளையில், 2500H என்பது நினைவக இருப்பிடத்தின் முகவரி. சிறும முக்கியத்துவம் வாய்ந்த பிட்டுகள் (00) L பதிவேட்டிலும், பெரும முக்கியத்துவம் வாய்ந்த பிட்டுகள் (25) H பதிவேட்டிலும் பளுவேற்ற வேண்டும்.

LDA addr

[A] ← [addr]

கட்டளையில் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய நினைவக இருப்பிடத்தில் உள்ளதைச் சேமக்கலத்தில் பளுவேற்ற வேண்டும் (உ.ம்) LDA 2400H. இது ஒரு 3–பைட் கட்டளையாகும். குறியீடு முறையில் இந்தக் கட்டளையை 3A, 00, 24 எனக் குறிப்பிடலாம். முதல் பைட் கட்டளையின் இயக்கக் குறியீடு (3A) இரண்டு மற்றும் மூன்றாவது பைட் நினைவக இருப்பிடத்தின் முகவரியைக் குறிக்கும்.

STA addr

[addr] ← [A]

சேமக்கலத்தில் உள்ளதைக் கட்டளையில் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய நினைவக இருப்பிடத்தில் சேமிக்க வேண்டும்.
(உ.ம்) STA 2000H

LHLD addr

[L] ← addr, [H] ← [addr + 1]

கட்டளையில் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய நினைவக இருப்பிடத்தில் உள்ளதை L பதிவேட்டில் பளுவேற்ற வேண்டும். அடுத்த நினைவக இருப்பிடத்தில் உள்ளதை H பதிவேட்டில் பளுவேற்ற வேண்டும். (உ.ம்) LHLD 2500 H. இந்தக் கட்டளைப்படி 2500Hல் உள்ளதை L பதிவேட்டிலும், 2501Hல் உள்ளதை H பதிவேட்டிலும் பளுவேற்ற வேண்டும்.

SHLD addr

[addr] ← [L], [addr+1] ← [H]

L பதிவேட்டில் உள்ளதைக் கட்டளையில் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய நினைவக இருப்பிடத்தில் சேமிக்க வேண்டும். H பதிவேட்டில் உள்ளதைக் கட்டளையில் குறிப்பிடப்பட்டுள்ள முகவரியின் அடுத்த முகவரியையுடைய நினைவக இருப்பிடத்தில் சேமிக்க வேண்டும். (உ.ம்) SHLD 2500H. இந்தக் கட்டளைப்படி L பதிவேட்டில் உள்ளதை 2500H நினைவக இருப்பிடத்திலும், H பதிவேட்டில் உள்ளதை 2501H நினைவக இருப்பிடத்திலும் சேமிக்க வேண்டும்.

LDAX rP

[A] ←[[rP]]

கட்டளையில் குறிப்பிடப்பட்டுள்ள பதிவேடு ஜோடியில் உள்ள முகவரியையுடைய நினைவக இருப்பிடத்தின் உள்ளடக்கத்தைச் சேமக்கலத்தில் சேமிக்க வேண்டும். (உ.ம்) LDAX B. B-C பதிவேடு ஜோடியில் உள்ள முகவரியையுடைய நினைவக இருப்பிடத்தின் உள்ளடக்கத்தைச் சேமக்கலத்தில் பளுவேற்ற வேண்டும். இந்தக் கட்டளை B-C மற்றும் D-E பதிவேடு ஜோடிகளுக்கு மட்டும் பயன்படுத்தப்படும்.

STA X rP

[[rP]] ← [A]

சேமக்கலத்தில் உள்ளதைக் கட்டளையில் குறிப்பிடப்பட்டுள்ள பதிவேடு ஜோடியில் உள்ள முகவரியையுடைய நினைவக இருப்பிடத்தில் சேமிக்க வேண்டும். (உ.ம்) STAX D. சேமக்கலத்தில் உள்ளதை D-E பதிவேடு ஜோடியில் உள்ள முகவரியையுடைய நினைவக இருப்பிடத்தில் சேமிக்க வேண்டும். இந்தக் கட்டளை B-C மற்றும் D-E பதிவேடு ஜோடிகளுக்கு மட்டும் பயன்படுத்தப்படும்.

XCHG

[H-L] ↔ [D-E]

H-L பதிவேடு ஜோடியின் உள்ளடக்கம் மற்றும் D-E பதிவேடு ஜோடியின் உள்ளடக்கம் ஆகியவை ஒன்றுக்கொன்று பரிமாற்றம் செய்தல் வேண்டும்.

கணிதத் தொகுப்பு

ADD r

[A] ← [A] + [r]

பதிவேடு r இல் உள்ளவற்றைச் சேமக்கலத்தில் உள்ளவற்றுடன் கூட்டி, முடிவைச் சேமக்கலத்தில் வைக்க வேண்டும்.

ADD M

[A] ← [A] + [[H-L]]

H-L பதிவேடு ஜோடியில் உள்ள முகவரியையுடைய நினைவக இருப்பிடத்தில் உள்ளவற்றைச் சேமக்கலத்தில் உள்ளவற்றுடன் கூட்டி, முடிவைச் சேமக்கலத்தில் வைக்க வேண்டும்.

ADC r

[A] ← [A] + [r] +[CS]

பதிவேடு r இல் உள்ளவற்றையும், சுமை நிலைப்பாட்டுக் கொடியில் உள்ளவற்றையும், சேமக்கலத்தில் உள்ளவற்றுடன் கூட்டி, முடிவைச் சேமக்கலத்தில் வைக்க வேண்டும்.

ADC M

[A] ← [A] + [[H-L]]+ [CS]

H-L பதிவேடு ஜோடியால் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய நினைவக இருப்பிடத்தில் உள்ளவற்றையும் மற்றும் சுமை நிலைப்பாட்டுக் கொடியில் உள்ளவற்றையும்,

சேமக்கலத்தில் உள்ளவற்றுடன் கூட்டி, முடிவைச் சேமக்கலத்தில் வைக்க வேண்டும்.

ADI data

[A] ← [A] + data

(உ.ம்) ADI 08. இந்தக் கட்டளையின்படி 08 என்ற எண்ணை உடனடியாகச் சேமக்கலத்தில் உள்ளவற்றுடன் கூட்டி, முடிவைச் சேமக்கலத்தில் வைக்க வேண்டும்.

ACI data

[A] ← [A] + data + [CS]

கட்டளையில் குறிப்பிடப்பட்டுள்ள தரவையும் மற்றும் சுமை நிலைப்பாட்டுக் கொடியில் உள்ளவற்றையும் சேமக்கலத்தில் உள்ளவற்றுடன் கூட்டி, முடிவைச் சேமக்கலத்தில் வைக்க வேண்டும்.

DAD rp

[H-L] ← [H-L] + [rP]

பதிவேடு ஜோடி rp இல் உள்ளவற்றை H-L பதிவேடு ஜோடியில் உள்ளவற்றுடன் கூட்டி, முடிவை H-L ஜோடியில் வைக்க வேண்டும்.

SUB r

[A] ← [A] - [r]

பதிவேடு rஇல் உள்ளவற்றைச் சேமக்கலத்தில் உள்ளவற்றுடன் கழித்து, முடிவைச் சேமக்கலத்தில் வைக்க வேண்டும்.

SUB M

[A] ← [A] - [[H-L]]

H-L பதிவேடு ஜோடியில் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய நினைவக இருப்பிடத்தில் உள்ளவற்றைச் சேமக்கலத்தில் உள்ளவற்றுடன் கழித்து, முடிவைச் சேமக்கலத்தில் வைக்க வேண்டும்.

SBB r

[A] ← [A] + [r] + [CS]

பதிவேடு rஇல் உள்ளவற்றைச் சுமையுடன் சேர்த்து, சேமக்கலத்தில் உள்ளவற்றுடன் கழித்து, முடிவைச் சேமக்கலத்தில் வைக்க வேண்டும்.

SUI data

[A] ← [A] – data

கட்டளையில் கொடுக்கப்பட்ட தரவைச் சேமக்கலத்தில் உள்ளவற்றுடன் கழித்து, முடிவைச் சேமக்கலத்தில் வைக்க வேண்டும்.

SBI data

[A] ← [A] - data - [CS]

கட்டளையில் கொடுக்கப்பட்ட தரவு மற்றும் சுமையைச் சேமக்கலத்தில் உள்ளவற்றுடன் கழித்து, முடிவைச் சேமக்கலத்தில் வைக்க வேண்டும்.

INR r

[r] ←[r] + 1

பதிவேடு r இல் உள்ளவற்றை ஒரு பிட் அதிகப்படுத்த வேண்டும்.

INR M

[[H-L] ← [[H-L] + 1

H-L பதிவேடு ஜோடியில் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய நினைவக இருப்பிடத்தில் உள்ளவற்றை ஒரு பிட் அதிகப்படுத்த வேண்டும்.

DCR r

[r] ← [r] -1 .

பதிவேடு r இல் உள்ளவற்றை ஒரு பிட் குறைக்க வேண்டும்.

DCR M

[[H-L] ← [[H-L]] – 1

H-L பதிவேடு ஜோடியில் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய நினைவக இருப்பிடத்தில் உள்ளவற்றை ஒரு பிட் குறைக்க வேண்டும்.

INX rp

[rP] ← [rP] + 1

பதிவேடு ஜோடி rp இல் உள்ளவற்றை ஒரு பிட் அதிகப்படுத்த வேண்டும்.

DCX rp

[rP] ← [rP]-1

பதிவேடு ஜோடி rpஇல் உள்ளவற்றை ஒரு பிட் குறைக்க வேண்டும்.

DAA

இந்தக் கட்டளை, திட்ட நிரலில் ADD, ADI, ACI மற்றும் ADC கட்டளைகளுக்குப் பிறகு பயன்படுத்தப்படும். ADD அல்லது ADC கட்டளைகளைச் செயல்படுத்திய பிறகு முடிவு பதினாறடி எண்ணாகச் சேமக்கலத்தில் இருக்கும். DAA கட்டளை சேமக்கலத்தில் உள்ள முடிவைப் பத்தடி முறைக்கு மாற்றும்.

வாதியல் தொகுப்பு

ANA r

[A] ← [A] ∧ [r]

பதிவேடு rஇல் உள்ளவை சேமக்கலத்தில் உள்ளவற்றுடன் உம் (AND) வினைப்பட்டு வருகின்ற முடிவைச் சேமக்கலத்தில் வைக்க வேண்டும்.

ANA M

[A] ← [A] ∧ [[H-L]]

H-L பதிவேடு ஜோடியில் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய நினைவக இருப்பிடத்தில் உள்ளவை சேமக்கலத்தில் உள்ளவற்றுடன் உம் வினைப்பட்டு வருகின்ற முடிவைச் சேமக்கலத்தில் வைக்க வேண்டும்.

ANI data

[A] ← [A] ∧ [data]

கட்டளையில் கொடுக்கப்பட்ட தரவு சேமக்கலத்தில் உள்ளவற்றுடன் உம் வினைப்பட்டு வருகின்ற முடிவைச் சேமக்கலத்தில் வைக்க வேண்டும்.

ORA M

[A] ← [A] ∨ [[H-L]]

H-L பதிவேடு ஜோடியில் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய நினைவக இருப்பிடத்தில் உள்ளவை சேமக்கலத்தில் உள்ளவற்றுடன் அல்ல வினைப்பட்டு வருகின்ற முடிவைச் சேமக்கலத்தில் வைக்க வேண்டும்.

ORA data

[A] ← [A] ∨ data

கட்டளையில் கொடுக்கப்பட்ட தரவு சேமக்கலத்தில் உள்ளவற்றுடன் அல்ல வினைப்பட்டு வருகின்ற முடிவைச் சேமக்கலத்தில் வைக்க வேண்டும்.

XRA r

[A] ←[A]∀[r]

பதிவேடு r இல் உள்ளவை சேமக்கலத்தில் உள்ளவற்றுடன் தவிர்க்கப்பட்ட அல்ல (EX-OR) வினைப்பட்டு வருகின்ற முடிவைச் சேமக்கலத்தில் வைக்க வேண்டும்.

XRA M

[A] ← [A] ∀ [[H-L]]

H-L பதிவேடு ஜோடியில் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய நினைவக இருப்பிடத்தில் உள்ளவை சேமக்கலத்தில் உள்ளவற்றுடன் தவிர்க்கப்பட்ட அல்ல வினைப்பட்டு வருகின்ற முடிவைச் சேமக்கலத்தில் வைக்க வேண்டும்.

XRI data

[A] ← [A] ∀ data

கட்டளையில் கொடுக்கப்பட்ட தரவு சேமக்கலத்தில் உள்ளவற்றுடன் தவிர்க்கப்பட்ட அல்ல வினைப்பட்டு வருகின்ற முடிவைச் சேமக்கலத்தில் வைக்க வேண்டும்.

CMA

சேமக்கலத்தில் உள்ளதை ஒன்றின் நிரப்பு முறையில் (I's compliment) மாற்றம் செய்து, முடிவை சேமக்கலத்தில் வைக்க வேண்டும். ஒரு ஈரடி எண்ணை ஒன்றின் நிரப்பு முறையில் மாற்றம் செய்யும்போது 0 என்றால் 1 எனவும், 1 என்றால் 0 எனவும் மாறும்.

CMC

சுமை நிலைப்பாட்டுக் கொடியில் உள்ளதை ஒன்றின் நிரப்பு முறையில் மாற்றம் செய்ய வேண்டும்.

STC

சுமை நிலைப்பாட்டுக் கொடியை ஒன்று என அமைவு செய்ய வேண்டும்.

CMP r

பதிவேடு rஇல் உள்ளதைச் சேமக்கலத்தில் உள்ளவற்றுடன் ஒப்பிட வேண்டும். அதாவது பதிவேடு rஇல் உள்ளதைச் சேமக்கலத்தில் உள்ளவற்றுடன் கழித்து, நிலைப்பாட்டுக் கொடியைக் கழித்தல் வினைக்குத் தகுந்தவாறு அமைவு செய்ய வேண்டும்.

CMP M

[A] – [[H-L]]

H-L பதிவேடு ஜோடியில் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய நினைவக இருப்பிடத்தில் உள்ளவற்றைச் சேமக்கலத்தில் உள்ளவற்றுடன் கழித்து, நிலைப்பாட்டுக் கொடியைக் கழித்தல் வினைக்குத் தகுந்தவாறு அமைவு செய்ய வேண்டும்.

CPI data

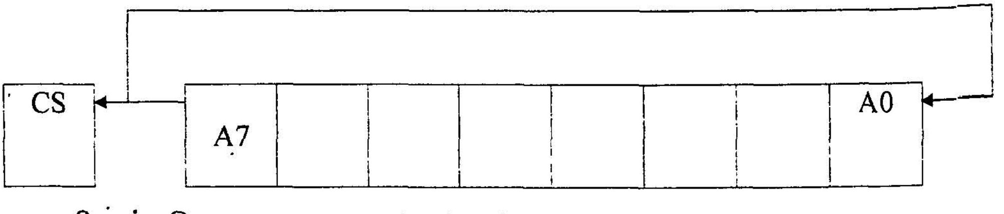
[A] - data

கட்டளையில் கொடுக்கப்பட்ட தரவைச் சேமக்கலத்தின் உள்ளவற்றுடன் கழித்து, நிலைப்பாட்டுக் கொடியைக் கழித்தல் வினைக்குத் தகுந்தவாறு அமைவு செய்ய வேண்டும்.

RLC

$[A_{n+1}] \leftarrow [A_n]$, $[A_o] \leftarrow [A_7]$,$[CS] \leftarrow [A_7]$

சேமக்கலத்தில் உள்ளதை இடது பக்கமாக ஒரு பிட் சுற்ற வேண்டும். சேமக்கலத்தின் ஏழாவது பிட்டை, சுமை பிட்டுக்கும் மற்றும் சுழி பிட்டுக்கும் நகர்த்த வேண்டும்.

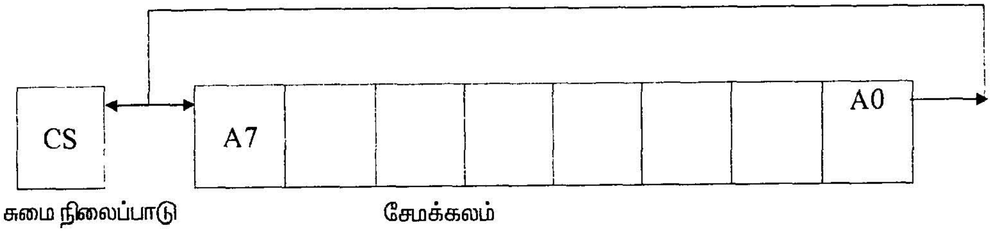


படம் (1.7)

RRC

$[A_7] \leftarrow [A_0]$, $[CS] \leftarrow [A_0]$, $[A_n] \leftarrow [A_{n+1}]$

சேமக்கலத்தில் உள்ளதை வலதுபக்கமாக ஒரு பிட் சுற்ற வேண்டும். சேமக்கலத்தின் சுழி பிட்டை, ஏழாவது பிட்டுக்கும் மற்றும் சுமை பிட்டுக்கும் நகர்த்த வேண்டும்.

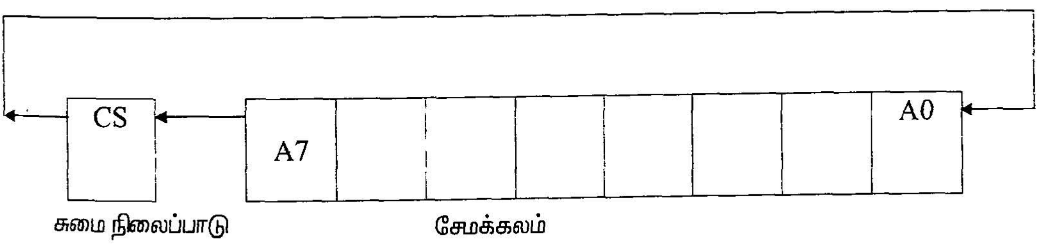


படம் (1.8)

RAL

$[A_{n+1}] \leftarrow [A_n]$, $[CS] \leftarrow [A_7]$, $[A_0] \leftarrow [CS]$

சேமக்கலத்தில் உள்ளதை இடது பக்கமாகச் சுமையோடு சேர்த்து ஒரு பிட் சுற்ற வேண்டும். சேமக்கலத்தின் ஏழாவது பிட் சுமைக்கும் மற்றும் சுமை பிட் சேமக்கலத்தின் சுழி பிட்டுக்கும் நகர்த்த வேண்டும்.

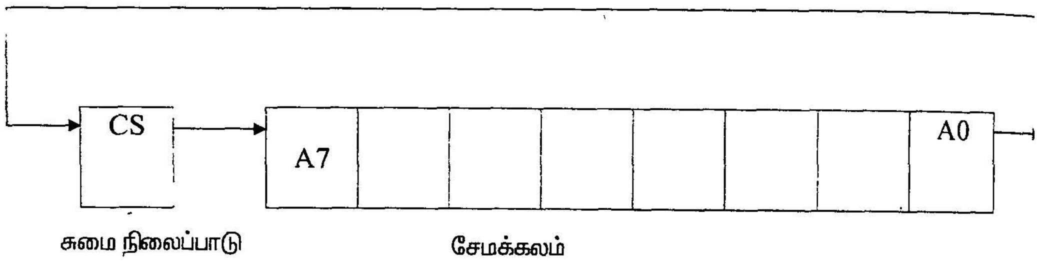


படம் (1.9)

RAR

$[A_n] \leftarrow [A_{n+1}]$, $[CS] \leftarrow [A_0]$, $[A_7] \leftarrow [CS]$

சேமக்கலத்தில் உள்ளதை வலது பக்கமாகச் சுமையோடு சேர்த்து ஒரு பிட் சுற்ற வேண்டும். சேமக்கலத்தின் சுழி பிட் சுமைக்கும் மற்றும் சுமை பிட் சேமக்கலத்தின் ஏழாவது பிட்டுக்கும் நகர்த்த வேண்டும்.



படம் (1.10)

கிளைத் தொகுப்பு

இந்தத் தொகுப்பில் உள்ள கட்டளைகள், திட்ட நிரலில் உள்ள கட்டளைகளின் இயல்பு வரிசையில் மாற்றம் செய்யும். கிளைத் தொகுப்பு கட்டளைகள் இரண்டு வகைப்படும். அவை நிபந்தனை மற்றும் நிபந்தனையற்ற கட்டளைகள். நிபந்தனை கிளைக் கட்டளைகள், சில நிபந்தனைகளைத் திருப்தி செய்யும்போது, இயல்பு வரிசையில் மாற்றம் ஏற்பட்டு, நிபந்தனை கட்டளையில் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய நினைவக இருப்பிடத்திலுள்ள கட்டளை செயல்படுத்தப்படும்.

நிபந்தனையற்ற கிளை கட்டளைகள் திட்ட நிரலில் வரும் போது, இயல்பான வரிசையில் மாற்றம் ஏற்பட்டு, நிபந்தனையற்ற கட்டளையில் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய நினைவக இருப்பிடத்தில் உள்ள கட்டளை செயல்படுத்தப்படும்.

நிபந்தனையற்ற கட்டளை

(உ.ம்) JMP addr. இந்தக் கட்டளை திட்ட நிரலில் வரும்போது, இயல்பான வரிசையில் மாற்றம் ஏற்பட்டு, கட்டளையில் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய நினைவக இருப்பிடத்தில் உள்ள கட்டளை செயல்படுத்தப்படும்.

நிபந்தனை கட்டளை

JZ addr

முந்தைய கட்டளையைப் பொறுத்து முடிவு சுழியாக இருந்தால், கட்டளையில் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய நினைவக இருப்பிடத்திலுள்ள கட்டளை செயல்படுத்தப்படும்.

JNZ addr

முந்தைய கட்டளையைப் பொறுத்து முடிவு சுழியாக இல்லாவிட்டால், கட்டளையில் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய நினைவக இருப்பிடத்திலுள்ள கட்டளை செயல்படுத்தப்படும்.

JC addr

முந்தைய கட்டளையைப் பொறுத்து முடிவில் சுமை வந்தால், கட்டளையில் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய நினைவக இருப்பிடத்திலுள்ள கட்டளை செயல்படுத்தப்படும்.

JNC addr

முந்தைய கட்டளையைப் பொறுத்து முடிவில் சுமை இல்லாவிட்டால், கட்டளையில் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய நினைவக இருப்பிடத்திலுள்ள கட்டளை செயல்படுத்தப்படும்.

JP addr

முந்தைய கட்டளையைப் பொறுத்து முடிவு நேர்க்குறியுடன் இருந்தால், கட்டளையில் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய நினைவக இருப்பிடத்திலுள்ள கட்டளை செயல்படுத்தப்படும்.

JM addr

முந்தைய கட்டளையைப் பொறுத்து முடிவு எதிர்க்குறியுடன் இருந்தால், கட்டளையில் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய நினைவக இருப்பிடத்திலுள்ள கட்டளை செயல்படுத்தப்படும்.

JPE addr

முந்தைய கட்டளையைப் பொறுத்து முடிவில் எண்ணிலக்க 1 இரட்டைப் படையாக இருந்தால், கட்டளையில் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய நினைவக இருப்பிடத்திலுள்ள கட்டளை செயல்படுத்தப்படும்.

JPO addr

முந்தைய கட்டளையைப் பொறுத்து முடிவில் எண்ணிலக்க 1 ஒற்றைப் படையாக இருந்தால், கட்டளையில் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய நினைவக இருப்பிடத்திலுள்ள கட்டளை செயல்படுத்தப்படும்.

அழைப்புக் கட்டளைகள் (Call instrucitons)

இந்தக் கட்டளைகள் துணை நிரற் செயல்முறையை (Subroutine program) அழைப்பதற்குப் பயன்படுத்தப்படுகின்றன. துணை நிரற்செயல்முறையைச் செயல்படுத்துவதற்கு முன், முதன்மைத் திட்ட நிரலிலுள்ள அடுத்த கட்டளையின் முகவரி, அடுக்கில் சேமித்து வைக்கப்படும். புதிய அடுக்கு உச்சியின் முகவரியை தெரியப்படுத்த, அடுக்குச் சுட்டிக் காட்டியின் உள்ளடக்கம் குறைக்கப்படும். பிறகு அழைப்புக் கட்டளையில் குறிப்பிடப்பட்டுள்ள முகவரியிலிருந்து தொடங்கக்கூடிய துணை நிரற் செயல்முறை செயல்படுத்தப்படும் இந்த அழைப்புக் கட்டளைகள் இரண்டு வகைப்படும். ஒன்று நிபந்தனையுடன் கூடிய அழைப்புக் கட்டளை. மற்றொன்று நிபந்தனையற்ற அழைப்புக் கட்டளை (உ.ம்) (Call addr).

நிபந்தனையுடன் கூடிய அழைப்புக் கட்டளை

CC addr

சுமை நிலைப்பாடு கொடி ஒன்றாக இருந்தால், துணை நிரற் செயல்முறை செயல்படுத்தப்படும்.

CNC addr

சுழி நிலைப்பாடு கொடி சுழியாக இருந்தால், துணை நிரற் செயல்முறை செயல்படுத்தப்படும்.

CP addr

குறியீடு நிலைப்பாடு கொடி சுழியாக இருந்தால், துணை நிரற் செயல்முறை செயல்படுத்தப்படும்.

CM addr

குறியீடு நிலைப்பாடு கொடி ஒன்றாக இருந்தால், துணை நிரற் செயல்முறை செயல்படுத்தப்படும்.

CPE addr

சமநிலை நிலைப்பாடு கொடி ஒன்றாக இருந்தால், துணை நிரற் செயல்முறை செயல்படுத்தப்படும்.

CPO addr

சமநிலை நிலைப்பாடு கொடி சுழியாக இருந்தால், துணை நிரற் செயல்முறை செயல்படுத்தப்படும்.

திருப்புக் கட்டளைகள் (Return Instruction)

இந்தக் கட்டளை துணை நிரற் செயல்முறையின் முடிவில் பயன்படுத்தப்படும். துணை நிரற் செயல்முறையைச் செயல்படுத்துவதற்கு முன், முதன்மைத் திட்ட நிரலிலுள்ள அடுத்த கட்டளையின் முகவரி அடுக்கில் சேமித்து வைக்கப்படும். RET கட்டளையைச் செயல்படுத்துவதால், அடுக்கில் சேமித்து வைக்கப்பட்ட முகவரி திட்டநிரல் எண்ணிக்கு மாற்றப்படும். RET கட்டளையுடன் துணை நிரற் செயல்முறையின் செயல்பாடு முடிந்துவிடும். பின் முதன்மைத் திட்ட நிரலிலுள்ள CALL கட்டளைக்கு அடுத்துள்ள கட்டளை செயல்படுத்தப்படும். இந்தத் திருப்புக் கட்டளைகள் இரண்டு வகைப்படும். ஒன்று நிபந்தனையுடன் கூடிய திருப்புக் கட்டளை. மற்றொன்று நிபந்தனையற்ற திருப்புக் கட்டளை. (உ.ம்) RET

நிபந்தனையுடன் கூடிய திருப்புக் கட்டளை

RC

சுமை நிலைப்பாடு கொடி ஒன்றாக இருந்தால், துணை நிரற்செயல்முறையின் செயல்பாடு முடிந்து, முதன்மைத் திட்ட நிரலுக்குத் திரும்ப வேண்டும்.

RNC

சுமை நிலைப்பாடு கொடி சுழியாக இருந்தால், துணை நிரற் செயல்முறையின் செயல்பாடு முடிந்து, முதன்மைத் திட்ட நிரலுக்குத் திரும்ப வேண்டும்.

RZ

சுழி நிலைப்பாடு கொடி ஒன்றாக இருந்தால், துணை நிரற் செயல்முறையின் செயல்பாடு முடிந்து, முதன்மைத் திட்ட நிரலுக்கு த்திரும்ப வேண்டும்.

RNZ

சுழி நிலைப்பாடு கொடி சுழியாக இருந்தால், துணை நிரற் செயல்முறையின் செயல்பாடு முடிந்து முதன்மைத் திட்ட நிரலுக்குத் திரும்ப வேண்டும்.

RP

குறியீடு நிலைப்பாடு கொடி சுழியாக இருந்தால், துணை நிரற் செயல்முறையின் செயல்பாடு முடிந்து, முதன்மைத் திட்ட நிரலுக்குத் திரும்ப வேண்டும்.

RM

குறியீடு நிலைப்பாடு கொடி ஒன்றாக இருந்தால் துணை நிரற் முறையின் செயல்பாடு முடிந்து, முதன்மைத் திட்ட நிரலுக்குத் திரும்ப வேண்டும்.

RPE

சமநிலை நிலைப்பாடு கொடி ஒன்றாக இருந்தால், துணை நிரற் செயல்முறையின் செயல்பாடு முடிந்து, முதன்மைத் திட்ட நிரலுக்குத் திரும்ப வேண்டும்.

RPO

சமநிலை நிலைப்பாடு கொடி சுழியாக இருந்தால், துணை நிரற் செயல்முறையின் செயல்பாடு முடிந்து, முதன்மைத் திட்ட நிரலுக்குத் திரும்ப வேண்டும்.

RST n

இது மறு ஆரம்பக் கட்டளையாகும். இதுவும் CALL கட்டளை வகையைச் சேர்ந்தது. இதை செயல்படுத்துவதற்கு முன், திட்ட நிரல் எண்ணியில் உள்ளதை அடுக்கில் சேமிக்க வேண்டும். பின் முதன்மைத் திட்ட நிரலைச் செயல்படுத்தாமல் மறு ஆரம்ப இருப்பிடங்களிலுள்ள கட்டளை செயல்படுத்தப்படும். மறு ஆரம்பக் கட்டளைகளும் மற்றும் அதன் இருப்பிடங்களும் கீழே கொடுக்கப்பட்டுள்ளன.

கட்டளை	இயக்கக் குறியீடு	மறு ஆரம்ப இருப்பிடங்கள்
RST0	C7	0000
RST1	CF	0008
RST2	D7	0010
RST3	DF	0018
RST4	E7	0020
RST5	EF	0028
RST6	F7	0030
RST7	FF	0038

PCHL

H-L பதிவேடு ஜோடியில் உள்ளவற்றைத் திட்டநிரல் எண்ணிக்கு மாற்ற வேண்டும்.

அடுக்கு உள்ளீடு வெளியீடு மற்றும் கருவி கட்டுப்பாட்டுத் தொகுப்பு

In port – address

கட்டளையில் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய உள்ளீடு நிலையத்தில் உள்ள தரவுகளை சேமக்கலத்திற்கு நகர்த்த வேண்டும். (உ.ம்) IN 01. இதில் 01 என்பது நிலையம் Bயின் முகவரியாகும். இதில் உள்ள தரவுகளைச் சேமக்கலத்திற்கு நகர்த்த வேண்டும்.

Out part – addres

சேமக்கலத்தில் உள்ளதைக் கட்டளையில் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய வெளியீடு நிலையத்திற்கு மாற்ற வேண்டும். (உ.ம்) OUT 00. இதில் 00 என்பது நிலையம் Aயின் முகவரியாகும். சேமக்கலத்தில் உள்ளதை நிலையம் Aக்கு நகர்த்த வேண்டும்.

PUSH

பதிவேடு ஜோடி rpயில் உள்ளதை அடுக்குக்கு நகர்த்த வேண்டும்.

PUSH PSW

சேமக்கலத்தில் உள்ளதை அடுக்குக்கு நகர்த்த வேண்டும். நிலைப்பாடு கொடிகளில் உள்ளதையும் அடுக்குக்கு நகர்த்த வேண்டும்.

POP rP

அடுக்கில் சேமித்து வைக்கப்பட்ட பதிவேடு ஜோடி rPயின் உள்ளடக்கத்தைத் திரும்பப் பதிவேடு ஜோடிக்கு நகர்த்த வேண்டும்.

POP PSW

அடுக்கில் சேமித்து வைக்கப்பட்ட முறைப்படுத்தி நிலைப்பாடு வார்த்தையின் (Processor status word) உள்ளடக்கத்தைத் திரும்ப முறைப்படுத்தி நிலைப்பாடு வார்த்தைக்கு நகர்த்த வேண்டும். அதேபோன்று அடுக்கில் சேமித்து வைக்கப்பட்ட சேமக்கலத்தின் உள்ளடக்கத்தைத் திரும்பச் சேமக்கலத்திற்கு நகர்த்த வேண்டும்.

HLT

இந்தக் கட்டளையின் மூலம் நுண்முறைப்படுத்தியின் செயல்பாடு நிறுத்தப்படும். பதிவேடு மற்றும் நிலைப்பாடு கொடிகளின் உள்ளடக்கம் மாற்றப்படாது.

XTHL

L பதிவேட்டில் உள்ளதை அடுக்கு உச்சியின் பைட்டுடன் பரிமாற்றம் செய்ய வேண்டும். H பதிவேட்டில் உள்ளதை அடுக்கு உச்சியின் கீழ் உள்ள பைட்டுடன் பரிமாற்றம் செய்ய வேண்டும்.

SPHL

H-L பதிவேடு ஜோடியில் உள்ளதை அடுக்குச் சுட்டிக்காட்டிக்கு நகர்த்த வேண்டும்.

EI

இந்தக் கட்டளை செயல்படுத்தும்போது தடங்கல் அமைப்பு செயலியக்கப்படுகிறது.

DI

இந்தக் கட்டளை செயல்படுத்தும்போது தடங்கல் அமைப்பு செயல் முடக்கம் செய்யப்படுகிறது.

SIM

இந்தக் கட்டளையைச் செயல்படுத்துவதால் சேமக்கலத்தின் ஏழாவது பிட்டில் உள்ளவை தொடர் வெளியீடு தரவு (Serial output data) கோட்டிற்கு மாற்றப்படுகிறது.

RIM

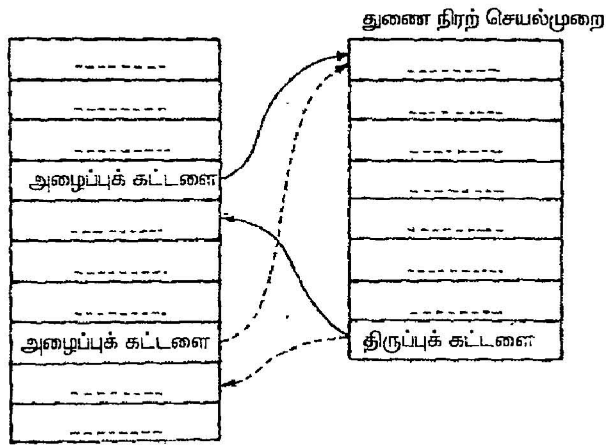
இந்தக் கட்டளையைச் செயல்படுத்துவதால் தொடர் உள்ளீடு தரவு (Serial input data) கோட்டிலுள்ள தரவு சேமக்கலத்தின் ஏழாவது பிட்டிற்கு மாற்றப்படுகிறது.

NOP

இந்தக் கட்டளையைச் செயல்படுத்துவதால் எந்த இயக்கமும் நடைபெறாது.

1.9 துணை நிரற் செயல்முறை (Subroutine)

திட்ட நிரல் எழுதும்போது சில இயக்கங்கள் அடிக்கடி வரலாம். இந்த இயக்கங்களுக்குத் தனிப்பட்ட கட்டளை கிடையாது. முதன்மைத் திட்ட நிரலில், அந்த இயக்கங்களுக்குத் திரும்பத் திரும்பத் திட்ட நிரல் எழுத நேரிடும். (உ.ம்) பெருக்கல் செயல். இன்டெல் 8085 நுண்முறைப்படுத்தியில் பெருக்கல் செயலுக்குத் தனிப்பட்ட கட்டளை கிடையாது. முதன்மைத் திட்ட நிரலில், சிறிய திட்ட நிரல்கள் அடிக்கடி வருவதைத் தடுப்பதே துணை நிரற் செயல்முறையின் நோக்கமாகும். ஒரு குறிப்பிட்ட செயலுக்கான சிறிய திட்ட நிரலே துணை நிரற் செயல்முறையாகும். இவை தனியாக எழுதப்பட்டு, நினைவகங்களில் சேமிக்கப்படும். இவற்றை முதன்மைத் திட்டநிரலைச் செயல்படுத்தும் போது பல இடங்களில் தேவைப்பட்டால் அழைப்புக் கட்டளை மூலம் பயன்படுத்திக் கொள்ளலாம். துணைச் செயல்முறையின் கடைசி கட்டளை திருப்புக் கட்டளையாகும். துணை நிரற் செயல்முறை செயல்படுத்தி முடிந்தவுடன், திரும்பவும் முதன்மைத் திட்ட நிரல், CALL கட்டளைக்கு அடுத்த கட்டளையிலிருந்து செயல்படுத்தப்படும். முதன்மைத் திட்ட நிரலில் ஒன்றுக்கும் மேற்பட்ட துணை நிரற் செயல்முறை இருக்கலாம். சில முக்கியமான துணை செயல்முறை ROMல் உற்பத்தியாளரால் சேமித்து வைக்கப்படும். படம் **(1.11)**ல் அழைப்புத் திருப்பு அமைப்பு காட்டப்பட்டுள்ளது.



படம் **(1.11)**

துணை நிரற் செயல்முறைக்கான அழைப்புத் திருப்பு அமைப்பு

1.10. இடைநிலை மொழியில் எழுதப்பட்ட திட்ட நிரலுக்கான சில உதாரணங்கள்

இரு 8 – பிட் எண்களின் கூட்டல் : கூட்டல் மதிப்பு 8 – பிட்டுகள்

நினைவக முகவரி	கருவிக் குறியீடு	செயற்குறிப்பு	இயமிகள்	விளக்கம்
2000	21,01,25	LXI	H, 2501H	முதல் எண்ணின் முகவரி, H-L பதிவேடு ஜோடிக்குக் கொண்டு வர வேண்டும்.
2003	7E	MOV	A,M	முதல் எண்ணைச் சேமக்கலத்திற்கு நகர்த்த வேண்டும்.
2004	23	INX	H	H-L பதிவேடு ஜோடியில் உள்ளதை அதிகரிக்க வேண்டும்.
2005	86	ADD	M	முதல் மற்றும் இரண்டாம் எண்ணைக் கூட்ட வேண்டும்
2006	32,03,25	STA	2503H	முடிவை நினைவக இருப்பிடம் 2503H இல் சேமிக்க வேண்டும்.
2009	76	HLT		திட்ட நிரல் செயல்பாடு நிறுத்தப்படும்.

(உ.ம்) தரவுகள் முடிவு

2501 – 49H 2503 – 9FH

2502 – 56H

இரு 8 – பிட் எண்களைக் கழித்தல்

நினைவக முகவரி	கருவிக் குறியீடு	செயற்குறிப்பு	இயமிகள்	விளக்கம்
2000	21,01,25	LXI	H, 2501H	முதல் எண்ணின் முகவரி, H-L பதிவேடு ஜோடிக்குக் கொண்டு வர வேண்டும்.
2003	7E	MOV	A,M	முதல் எண்ணைச் சேமக்கலத்திற்கு நகர்த்த வேண்டும்.
2004	23	INX	H	H-L பதிவேடு ஜோடியின் உள்ளடக்கத்தை 2501லிருந்து 2502H ஆக அதிகரிக்க வேண்டும்.
2005	96	SUB	M	இரண்டாம் எண்ணை முதல் எண்ணிலிருந்து கழிக்க வேண்டும்.
2006	23	INX	H	H – L பதிவேடு ஜோடியின் உள்ளடக்கத்தை அதிகரிக்க வேண்டும்.
2007	77	MOV	M,A	முடிவை 2503Hஇல் சேமிக்க வேண்டும்.
2008	76	HLT		திட்ட நிரல் செயல்பாடு நிறுத்தப்படும்.

(உ.ம்) தரவுகள்
2501 – 49H
2502 – 32H

முடிவு
2503 – 17H

தரவுகளின் வரிசையில் பெரிய எண் கண்டுபிடித்தல்

நினைவக முகவரி	கருவிக் குறியீடு	லேபில்	செயற்குறிப்பு	இயமிகள்	விளக்கம்
2000	21,00,25		LXI	H, 2500 H	எண்ணிக்கையின் முகவரி, H-L பதிவேடு ஜோடியில் இருக்க வேண்டும்.
2003	4E		MOV	C,M	எண்ணிக்கையைப் பதிவேடு Cக்கு நகர்த்த வேண்டும்.
2004	23		INX	H	முதல் எண்ணின் முகவரி H-L பதிவேடு ஜோடியில் இருக்க வேண்டும்.
2005	7E		MOV	A,M	முதல் எண் சேமக்கலத்தில் இருக்க வேண்டும்.
2006	0D		DCR	C	எண்ணிக்கையைக் குறைக்க வேண்டும்
2007	23	LOOP	INX	H	H-L பதிவேடு ஜோடியின் உள்ளடக்கத்தை அதிகப்படுத்தப் வேண்டும்.
2008	BE		CMP	M	நினைவகத்திலுள்ள எண்ணை முந்தைய எண்ணுடன் ஒப்பிட வேண்டும்.
2009	D2, 0D, 20		JNC	AHEAD	நினைவகத்திலுள்ள எண் முந்தைய எண்ணுடன் பெரியதா ? இல்லை. பெரிய எண் சேமக்கலத்தில் இருக்கிறது. லேபில் AHEADக்குச் செல்ல வேண்டும்.

200C	7E		MOV	A,M	ஆம், பெரிய எண் சேமக்கலத்திற்கு வர வேண்டும்.
200D	0D	AHEAD	DCR	C	எண்ணிக்கையைக் குறைக்க வேண்டும்.
200E	C2, 07, 20		JNZ	LOOP	எண்ணிக்கை சுழியாக இல்லாவிட்டால் லேபில் LOOPக்குச் செல்ல வேண்டும்.
2011	32,50,24		STA	2450H	முடிவு 2450Hஇல் சேமிக்க வேண்டும்
2014	76		HLT	2450H	திட்ட நிரல் செயல்பாடு நிறுத்தப்படும்

(உ.ம்) தரவுகள்	முடிவு
2500 – 03	2450 – 99
2501 – 98	
2502 – 75	
2503 - 99	

தரவுகளின் வரிசையில் சிறிய எண்ணைக் கண்டுபிடித்தல்

நினைவக முகவரி	கருவிக் குறியீடு	லேபில்	செயற்குறிப்பு	இயமிகள்	விளக்கம்
2000	21,00,25		LXI	H, 2500 H	எண்ணிக்கையின் முகவரி, H-L பதிவேடு ஜோடியில் இருக்க வேண்டும்.
2003	4E		MOV	C,M	எண்ணிக்கையைப் பதிவேடு Cக்கு நகர்த்த வேண்டும்.
2004	23		INX	H	முதல் எண்ணின் முகவரி H-L பதிவேடு ஜோடியில் இருக்க வேண்டும்.
2005	7E		MOV	A,M	முதல் எண் சேமக்கலத்தில் இருக்க வேண்டும்.
2006	0D		DCR	C	எண்ணிக்கையைக் குறைக்க வேண்டும்
2007	23	LOOP	INX	H	H-L பதிவேடு ஜோடியின் உள்ளடக்கத்தை அதிகப்படுத்த வேண்டும்.
2008	BE		CMP	M	நினைவகத்திலுள்ள எண்ணை முந்தைய எண்ணுடன் ஒப்பிட வேண்டும்.
2009	DA, 0D, 20		JC	AHEAD	நினைவகத்திலுள்ள எண் முந்தைய எண்ணுடன் சிறியதா ? ஆம். சிறிய எண் சேமக்கலத்தில் இருக்க வேண்டும். லேபில் AHEADக்குச் செல்ல வேண்டும்.

200C	7E		MOV	A,M	இல்லை. அடுத்த எண் சேமக்கலத்திற்கு வர வேண்டும்.
200D	0D	AHEAD	DCR	C	எண்ணிக்கையைக் குறைக்க வேண்டும்.
200E	C2, 07, 20		JNZ	LOOP	எண்ணிக்கை சுழியாக இல்லாவிட்டால் லேபில் LOOPக்குச் செல்ல வேண்டும்.
2011	32,50,24		STA	2450H	சிறிய எண்ணை 2450Hஇல் சேமிக்க வேண்டும்
2014	76		HLT	2450H	திட்ட நிரல் செயல்பாடு நிறுத்தப்படும்

(உ.ம்) தரவுகள்	முடிவு
2500 – 03	2450 – 58H
2501 – 86H	
2502 – 58H	
2503 – 75H	

இரண்டு 8–பிட் எண்களைக் கூட்டுதல் : முடிவு 16–பிட்

நினைவக முகவரி	கருவிக் குறியீடு	லேபில்	செயற்குறிப்பு	இயமிகள்	விளக்கம்
2000	21,01,25		LXI	H,2501H	முதல் எண்ணின் முகவரி H-L பதிவேடு ஜோடியில் இருக்கும்.
2003	0E, 00		MVI	C, 00	கூட்டுத் தொகையின் MSB, பதிவேடு C இல் இருக்கும். C இன் ஆரம்ப மதிப்பு = 00
2005	7E		MOV	A,M	முதல் எண்ணைச் சேமக்கலத்திற்கு நகர்த்த வேண்டும்.
2006	23		INX	H	இரண்டாவது எண்ணின் முகவரி H-L – பதிவேடு ஜோடியில் இருக்கும்.
2007	86		ADD	M	முதல் மற்றும் இரண்டாவது எண்ணைக் கூட்ட வேண்டும்.
2008	D2,0C,20		JNC	AHEAD	கூட்டுத் தொகையில் சுமை வருகிறதா ? இல்லை என்றால் லேபில் AHEADக்குச் செல்ல வேண்டும்.
200B	0C		INR	C	சுமை வந்தால் Cஇன் உள்ளடக்கத்தை 1-பிட் அதிகரிக்க வேண்டும்.

200C	02,03,25	AHEAD	STA	2503H	கூட்டுத் தொகையின் LSBஐ 2503H என்ற நினைவக இருப்பிடத்தில் சேமிக்க வேண்டும்.
200F	79		MOV	A, C	கூட்டுத் தொகையின் MSBஐ சேமக்கலத்திற்கு நகர்த்த வேண்டும்.
2010	32,04,25		STA	2504H	MSBஐ 2504H என்ற நினைவக இருப்பிடத்தில் சேமிக்க வேண்டும்.
2013	76		HLT		திட்டநிரல் செயல்பாடு நிறுத்தப்படும்.

(உ.ம்) தரவுகள் முடிவு

2501 – 98H 2503 – 32H (LSB)

2502 - 9AH 2504 – 01H (MSB)

இரண்டு 8–பிட் எண்களைப் பெருக்குதல் : பெருக்கல் மதிப்பு 16–பிட்

நினைவக முகவரி	கருவிக் குறியீடு	லேபில்	செயற்குறிப்பு	இயமிகள்	விளக்கம்
2000	2A,01,25		LHLD	2501H	பெருக்கப்பட வேண்டிய எண் H-L பதிவேட்டு ஜோடியில் இருக்க வேண்டும்.
2003	EB		XCHG		பெருக்கப்பட வேண்டிய எண் D-E பதிவேட்டு ஜோடியில் இருக்கும்.
2004	3A,03,25		LDA	2503H	பெருக்கும் எண் சேமக்கலத்தில் இருக்கும்.
2007	21,00,00		LXI	H,0000	பெருக்கல் மதிப்பின் தொடக்க மதிப்பு அதாவது 00, H-L ஜோடியில் இருக்கும்.
200A	0E,08		MVI	C,08	எண்ணிக்கை = 8 என்பது பதிவேடு Cஇல் இருக்கும்.
200C	29	LOOP	DAD	H	கிடைக்கக் கூடிய பெருக்கல் மதிப்பை ஒரு பிட் இடது பக்கமாக நகர்த்த வேண்டும்.
200D	17		RAL		பெருக்கும் எண்ணை இடதுபக்கமாக ஒரு பிட் நகர்த்த வேண்டும். பெருக்கு எண்ணில் சுமை வருகிறதா ?
200E	D2,12,20		JNC	AHEAD	இல்லை. AHEAD என்ற லேபிலுக்குச் செல்ல வேண்டும்.
2011	19		DAD	D	பெருக்கல் மதிப்பு = பெருக்கல் மதிப்பு + பெருக்கப்பட வேண்டிய எண்

2012	0D	AHEAD	DCR	C	எண்ணிக்கையை ஒரு பிட் குறைக்க வேண்டும்.
2013	C2,0C,20		JNZ	LOOP	எண்ணிக்கைச் சுழியாக இல்லாவிட்டால் LOOP என்ற லேபிலுக்குச் செல்ல வேண்டும்.
2016	22,04,25		SHLD	2504	எண்ணிக்கைச் சுழியாக இருந்தால் முடிவைச் சேமிக்க வேண்டும்.
2019	76		HLT		திட்டநிரல் செயல்பாடு நிறுத்தப்படும்.

(உ.ம்)

தரவுகள்

2501 = 84H ⟶ பெருக்கப்பட வேண்டிய எண்ணின் சிறும முக்கியத்துவம் வாய்ந்த பிட்டுகள்

2502 = 00 ⟶ பெருக்கப்பட வேண்டிய எண்ணின் பெரும முக்கியத்துவம் வாய்ந்த பிட்டுகள்.

2503 = 56H ⟶ பெருக்கும் எண்.

முடிவு

2504 = 58H ⟶ பெருக்கல் மதிப்பின் சிறும முக்கியத்துவம் வாய்ந்த பிட்டுகள்.

2505 = 2C ⟶ பெருக்கல் மதிப்பின் பெரும முக்கியத்துவம் வாய்ந்த பிட்டுகள்.

8–பிட் வகுத்தல்

நினைவக முகவரி	கருவிக் குறியீடு	லேபில்	செயற்குறிப்பு	இயமிகள்	விளக்கம்
2400	2A,01,25		LHLD	2501H	வகுக்கப்படும் எண் H-L ஜோடியில் இருக்கும்.
2403	3A,03,25		LDA	2503H	2503H இல் உள்ள வகுக்கும் எண்ணைச் சேமக்கலத்தில் வைக்க வேண்டும்.
2406	47		MOV	B,A	சேமக்கலத்திலுள்ள வகுக்கும் எண்ணை B பதிவேட்டிற்கு நகர்த்த வேண்டும்.
2407	0E,08		MVI	C,08	எண்ணிக்கை = 8 என்பது C பதிவேட்டில் இருக்கும்.
2409	29	LOOP	DAD	H	வகுக்கப்படும் எண் மற்றும் ஈவை ஒரு பிட் இடது பக்கமாக நகர்த்த வேண்டும்.
240A	7C		MOV	A,H	வகுக்கப்படும் எண்ணின் பெரும முக்கியத்துவம் வாய்ந்த பிட்டுகளைச் சேமக்கலத்திற்கு நகர்த்த வேண்டும்,
240B	90		SUB	B	வகுக்கும் எண்ணைச் சேமக்கலத்திலுள்ள பிட்டுகளுடன் கழிக்க வேண்டும்.
240C	DA,11,24		JC	AHEAD	வகுக்கப்படும் எண்ணின் பெரும முக்கியத்துவம் வாய்ந்த பிட்டுகள் வகுக்கும் எண்ணை விடப் பெரியதா? இல்லை. AHEADக்குச் செல்ல வேண்டும்.

240F	67		MOV	H,A	வகுக்கப்படும் எண்ணின் பெரும முக்கியத்துவம் வாய்ந்த பிட்டுகள் H பதிவேட்டில் இருக்கும்.
2410	2C		INR	L	ஆம். ஈவுடன் ஒன்றைக் கூட்ட வேண்டும்.
2411	0D	AHEAD	DCR	C	எண்ணிக்கையை ஒரு பிட் குறைக்க வேண்டும்.
2412	C2,09,24		JNZ	LOOP	எண்ணிக்கையின் மதிப்பு சுழியா? இல்லையென்றால் LOOP என்ற லேபிலுக்குச் செல்ல வேண்டும்.
2415	22,04,25		SHLD	2504H	ஈவை **2504** என்ற நினைவக இருப்பிடத்திலும் மீதியை **2505**Hலும் சேமிக்க வேண்டும்.
2418	76		HLT		திட்டநிரல் செயல்பாடு நிறுத்தப்படும்.

(உ.ம்)

தரவுகள்

2501 – 9BH ⟶ வகுக்கப்படும் எண்ணின் சிறும முக்கியத்துவம் வாய்ந்த பிட்டுகள்

2502 – **48**H ⟶ வகுக்கப்படும் எண்ணின் பெரும முக்கியத்துவம் வாய்ந்த பிட்டுகள்

2503 – **1**AH ⟶ வகுக்கும் எண்

முடிவு

2504 – F**2** ⟶ ஈவு

2505 – **07** ⟶ மீதி

ஒரு 8–பிட் எண்ணின் இரண்டின் நிரப்பை கண்டுபிடித்தல்

நினைவக முகவரி	கருவிக் குறியீடு	செயற்குறிப்பு	இயமிகள்	விளக்கம்
2000	3A,01,25	LDA	2501H	2501H இல் உள்ள தரவுகளைச் சேமக்கலத்தில் வைக்க வேண்டும்.
2003	2F	CMA		சேமக்கலத்தில் உள்ள எண்ணுக்கு ஒன்றின் நிரப்பு காணுதல்
2004	3C	INR	A	இரண்டின் நிரப்பு காணுதல்
2005	32,02,25	STA	2502H	முடிவை 2502Hல் சேமிக்க வேண்டும்.
2008	76	HLT		திட்டநிரல் செயல்பாடு நிறுத்தப்படும்.

(உ.ம்)

தரவு	முடிவு
2501 - 96H	2502-6AH

8 – பிட் எண்ணை இடது பக்கமாக ஒரு பிட் நகர்த்துதல்

நினைவக முகவரி	கருவிக் குறியீடு	செயற்குறிப்பு	இயமிகள்	விளக்கம்
2000	3A,01,25	LDA	2501H	2501Hஇல் உள்ள தரவைச் சேமக்கலத்தில் வைக்க வேண்டும்.
2003	87	ADD	A	தரவை இடது பக்கமாக ஒரு பிட் நகர்த்துதல்
2004	32,02,25	STA	2502H	முடிவை 2502Hஇல் சேமிக்க வேண்டும்
2008	76	HLT		திட்டநிரல் செயல்பாடு நிறுத்தப்படும்.

(உ.ம்)

தரவு

2501 – 65H

முடிவு

2502 – CAH

ஒரு 8–பிட் எண்ணில் சிறும முக்கியத்துவம் வாய்ந்த நான்கு பிட்டுகளை மறைத்தல்

நினைவக முகவரி	கருவிக் குறியீடு	செயற்குறிப்பு	இயமிகள்	விளக்கம்
2000	3A,01,25	LDA	2501H	2501Hஇல் உள்ள தரவைச் சேமக்கலத்தில் வைக்க வேண்டும்.
2003	E6,F0	ANI	F0	சிறும முக்கியத்துவம் வாய்ந்த நான்கு பிட்டுகளை மறைத்தல்
2005	32,02,25	STA	2502H	முடிவை 2502Hஇல் சேமிக்க வேண்டும்
2008	76	HLT		திட்டநிரல் செயல்பாடு நிறுத்தப்படும்.

(உ.ம்)

தரவு முடிவு

2501 – A6 2502-A0

பயிற்சிகள்

1. மையச் செயலகத்தின் முக்கியமான பாகங்கள் யாவை? அவற்றின் செயல்பாடுகளைப் பற்றி விளக்குக.
2. உள்ளீடு சாதனத்தின் வேலை என்ன? அவற்றிற்குச் சில உதாரணங்கள் தருக.
3. வெளியீடு சாதனத்தின் வேலை என்ன? அவற்றிற்குச் சில உதாரணங்கள் தருக.
4. குறைக்கடத்தி நினைவகங்களைப் பற்றி விரிவாக எழுதுக.
5. 8085 நுண்முறைப்படுத்தியின் கணித மற்றும் வாதியல் அலகைப் பற்றி விளக்குக.
6. 8085ன் பதிவேடுகள் யாவை? அவற்றின் செயல்பாடுகளை விவரி.
7. பின்வரும் 8085ன் சைகைகளை விளக்குக.
   i) ALE ii) READY iii) IO/$\overline{M}$ iv) HOLD
   v) INTR vi) $\overline{WR}$ vii) S_0 மற்றும் S_1
8. இன்டெல் 8085இன் முகவரியிடல் முறைகளின் வகைகளைத் தகுந்த உதாரணத்துடன் விளக்குக.
9. 8085ன் தரவு மாற்றுத் தொகுப்பு மற்றும் கிளைத் தொகுப்புக் கட்டளைகளை விளக்குக.
10. 8085 கட்டளைகளின் அளவு பற்றிச் சிறுகுறிப்பு வரைக.
11. இயக்கக் குறியீடு மற்றும் இயமி என்றால் என்ன? உதாரணத்துடன் விளக்குக.
12. இன்டெல் 8085ன் நிலைப்பாட்டு கொடிகளை விளக்குக.
13. இரு 8–பிட் எண்களை கூட்டுவதற்கான இடைநிலை மொழித் திட்டநிரலை எழுதுக.
14. தரவுகளின் வரிசையில் சிறிய எண்ணைக் கண்டுபிடிப்பதற்கான திட்டநிரலை எழுதுக.
15. இரு 8 – பிட் எண்ணைக் கழிப்பதற்கானத் திட்டநிரலை எழுதுக.

2
இடைப் பரிமாற்று நினைவகம் மற்றும் உள்ளீடு / வெளியீடு கருவிகள்

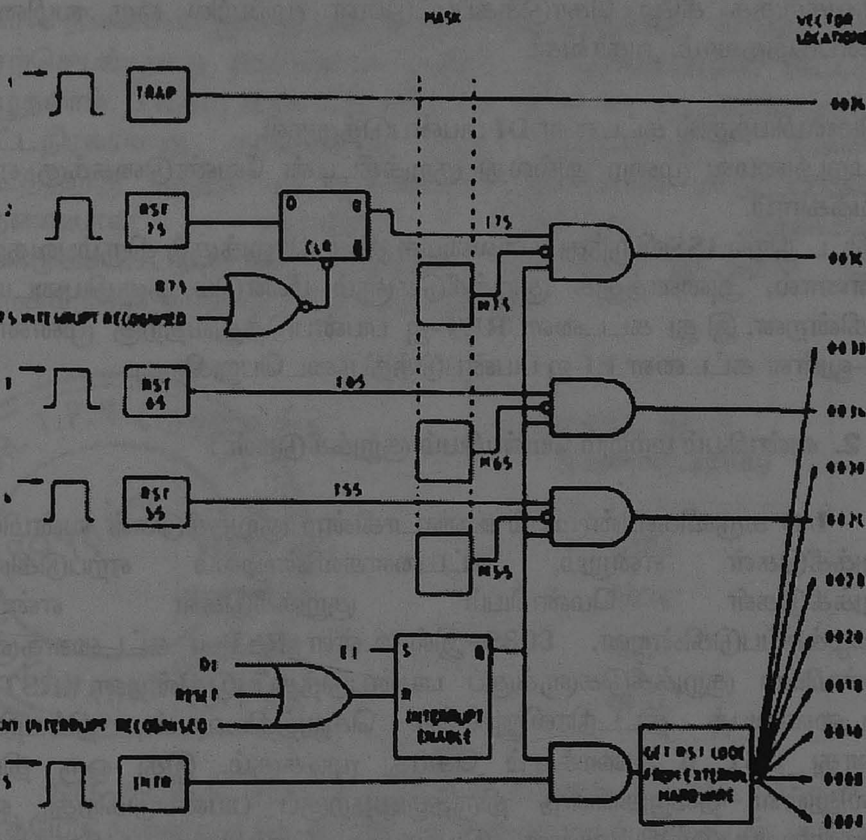
(Interfacing Memory and Input/Output Device)

2.1. 8085இன் குறுக்கீடுகள் :

பொதுவாகக் குறுக்கீடுகள், வன்மியம் குறுக்கீடுகள் மற்றும் மென்மியம் குறுக்கீடுகள் என்று இரண்டு பிரிவுகளாகப் பிரிக்கப்படுகின்றன. வெளியிலிருந்து நுண்முறைப்படுத்தியுடன் இணைக்கப்பட்டுள்ள புறவியல் சுற்றல் கருவிகளின் **(Peripheral Device)** மூலம் குறுக்கீடு முனைகளின் வழியே ஏற்படுத்தப்படுகின்ற குறுக்கீடுகள் வன்மியம் குறுக்கீடுகள் எனப்படும். கட்டளைகளின் மூலம் ஏற்படுகின்ற குறுக்கீடுகள் மென்மியம் குறுக்கீடுகள் எனப்படும். வன்மியம் குறுக்கீடுகள் முந்துரிமையின் அடிப்படையில் செயலாற்றும் தன்மை கொண்டதாகும். எனவே, இவைகள் முந்துரிமை குறுக்கீடுகள் எனவும் அழைக்கப்படும்.

8085 நுண்முறைப்படுத்தி **TRAP, RST 7.5, RST 6.5, RST 5.5** மற்றும் **INTR** என்று ஐந்து வகையான வன்மியம் குறுக்கீடுகளைக் கொண்டது. **TRAP** முதன்மை முந்துரிமைக் கொண்டதாகும். இதனை அடுத்து **RST 7.5, RST 6.5, RST 5.5** மற்றும் **INTR** என்கிற குறுக்கீடுகள் வரிசையான முந்துரிமைகளைக் கொண்டிருக்கும். குறுக்கீடுகள் பயன்படுத்தப்பட வேண்டுமென்றால், அவைகள் மென்மியத்தில் உள்ள தலைமைத் திட்டநிரலில் பயன்படுகின்ற கட்டளை **EI** ஐ செயலியக்கக் குறுக்கீடு **(Enable Interrupt)** மூலம் செயலியக்கப்பட வேண்டும். 8 085–இன் குறுக்கீடுகளின் சுற்று ருவ வரைபடம் 2.1–இல் காட்டப்பட்டுள்ளது. கட்டளை **EI,** குறுக்கீடுகள் செயலியக்கம் பெறுவதற்காகக் குறுக்கீடு செயலியக்க நிலைமாறியை அமைவு நிலையில் வைக்கிறது. அனைத்துக் குறுக்கீடுகளும் செயலியக்கம் அடைவதற்குக் கட்டளை **EI** பயன்படுத்தப்படுகிறது. குறுக்கீடுகள் செயல் முடக்கத்திற்கு கட்டளை **DI** செயல்முடக்கக் குறுக்கீடு **(Disable Interrupt)** பயன்படுகிறது. சில சூழமைவுகளில், நுண்முறைப்படுத்தியில் ஒரு குறிப்பிட்ட கட்டளைப் பணி நிகழ்த்தப்படும்பொழுது, குறுக்கீடுகளின் நிகழ்ச்சிகள் தடுக்கப்படுகின்றன. இது **DI** கட்டளையின் மூலம் நடத்தப்படுகிறது. கட்டளை **DI** குறுக்கீடு செயலியக்க நிலைமாறியை மறு அமைவுக்கும், செயல் முடக்கம் அடைவதற்காக

மறைக்கப்படாத குறுக்கீடு **TRAP** தவிர மற்ற குறுக்கீடுகள் அனைத்தையும் தகுதியற்றனவாகவும் மாற்றுகிறது. மறுஅமைவு **(RESET)** முறையும் முந்துரிமை செயலியக்க நிலைமாறியைச் செயல்முடக்கம் செய்கிறது.



படம் 2.1 8085 இன் குறுக்கீடுகளின் சுற்றுருவப் படம்

குறுக்கீடு வரி உயர்நிலைக்குச் செல்லும்போது முறைப்படுத்தி அதனுடைய தற்போதைய கட்டளையை முடிக்கிறது. மற்றும் அடுக்கில் திட்டநிரல் எண்ணியைச் சேமிக்கிறது. மேலும் **ISS**-ஐ எடுப்பதற்கு முன்பாகக் குறுக்கீடு செயலியக்க நிலைமாறியைத் தகுதியற்றதாக்குகிறது. அதன் காரணமாக மற்ற கருவிகள் மூலம்

நிகழ்த்தப்படுகின்ற குறுக்கீடுகள் ISS-இன் நிறைவேற்றத்தின்போது தவிர்க்கப்படுகின்றன. குறுக்கீடு செயலியக்க நிலைமாறியை மறுஅமைவில் வைப்பதின்மூலம் TRAP-ஐ தவிர அனைத்துக் குறுக்கீடுகளும் தகுதியற்றனவாக மாறுகின்றன.

இந்த நிலைமாறியை மறு அமைவுக்குக் கொண்டு வருவதற்குக் கீழே கொடுக்கப்பட்டுள்ள மூன்றில் ஒரு வழியைப் பயன்படுத்தலாம். அதாவது,

1) மென்மியத்தில் கட்டளை DI பயன்படுத்துதல்
2) மறுஅமைவு முறை அல்லது குறுக்கீட்டின் வேண்டுகைக்கு ஏற்ற அங்கீகாரம்.
3) திட்ட நிரல் ISSலிருந்து தலைமைத் திட்டநிரலுக்குத் திரும்புவதற்கு முன்னால், அனைத்துக் குறுக்கீடுகளும் மீண்டும் தகுதியுடையன வாகின்றன. இது கட்டளை RET-ஐ பயன்படுத்துவதற்கு முன்னால் ISS-லுள்ள கட்டளை EI-ஐ பயன்படுத்தி நடைபெறுகிறது.

2.1. 2. வன்மியம் மற்றும் மென்மியம் குறுக்கீடுகள் :

I/O கருவிகளின்மூலம் உண்டாகின்ற குறுக்கீடுகள் வன்மியம் குறுக்கீடுகள் என்றும், கட்டளைகளின்மூலம் ஏற்படுகின்ற குறுக்கீடுகள் மென்மியம் குறுக்கீடுகள் என்றும் அழைக்கப்படுகின்றன. 8085–இல் உள்ள RST n கட்டளைகளும் மென்மியம் குறுக்கீடுகளுக்குப் பயன்படுத்தப்படுகின்றன. RST n கட்டளையைத் திட்டநிரலினுள்ளே செலுத்தியவுடன், திட்டநிரல் ஆனது RST n வரைக்கும் செய்து முடிக்கும். இது ஒரு திட்ட நிரலிலுள்ள தவறுகளைத் திருத்துவதற்குப் பயன்படுகிறது. ஒரு நுண்முறைப்படுத்தியிலுள்ள இயல்பான செய்முறை வரிசையைத் தடுக்கும் உட்புற அசாதாரணமான அல்லது வழக்கமில்லாத நிபந்தனை விதிவிலக்காகும். உதாரணமாக, பூஜ்யத்தால் வகு என்பது விதிவிலக்கிற்குக் காரணமாகும். இன்டெல் இலக்கியம் விதிவிலக்கைப் பயன்படுத்துவதில்லை. ஆனால், மோட்ரோலா இலக்கியம் விதிவிலக்கைப் பயன்படுத்துகிறது. இன்டெல் மென்மியம், குறுக்கீடில் விதிவிலக்கைச் சேர்த்துக் கொள்கிறது.

2.1. 3. குறுக்கீடுகளின் அழைப்பு இடங்கள் :

குறுக்கீடு நேரிடும்போது, திட்டநிரல் ஒரு பிரத்யேகமான நினைவக இடத்திற்கு மாற்றப்படுகிறது. அதன்பிறகு மானிட்டர் பிரத்யேகமான நினைவக இடத்திலுள்ள திட்டநிரலை **RAM**-இல் உள்ள நினைவக இடத்திற்கு மாற்றுகிறது. குறுக்கீடு சேவை துணை நிரற்செயல்முறை **(ISS)**க்காக அதில் உபயோகிப்பாளர், திட்டநிரல் எழுதலாம். **TRAP, RST 7.5, RST 6.5, RST 5.5** குறுக்கீடுகளுக்குத் திட்டநிரலானது தானாகவே ஒரு பிரத்யேகமான நினைவக இடத்திற்கு எந்தப் புற வன்மியமும் இல்லாமல மாற்றப்படுகிறது. தேவையான வன்மியம் **8085**இல் முன்பாகவே கொடுக்கப்பட்டுள்ளது. பிரத்யேக நினைவக இடத்தில் உள்ள குறுக்கீடுகள் கீழே கொடுக்கப்பட்டுள்ளன.

DR. T.P.M. LIBRARY
MADURAI KAMARAJ UNIVERSITY
279067

குறுக்கீடு	அழைப்பு இடம் (பதினாறடியில்)
TRAP	**0024**
RST 7.5	**003C**
RST 6.5	**0034**
RST 5.5	**002C**

ஒரு குறுக்கீட்டிற்காக வன்மியம் தானாகவே திட்டநிரலை ஒரு பிரத்யேக நினைவக இடத்திற்கு மாற்றுவதற்குத் திசையன் குறுக்கீடு **(Vectored Interrupt)** என்று பெயர்.

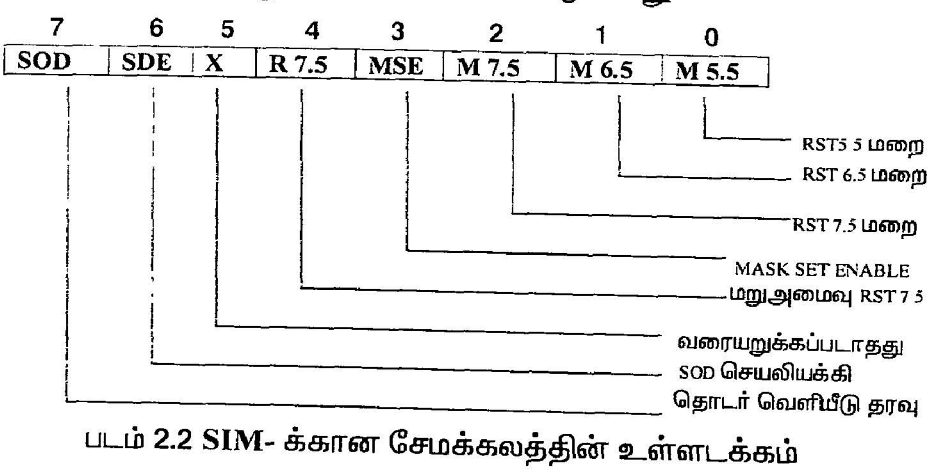
INTR அழைப்பு இடங்கள் :

INTR குறுக்கீடு **8** அழைப்பு இடங்களைக் கொண்டது. அட்டவணை **2.1**இல் அழைப்பு இடங்கள், **RST n** கட்டளைகள் மற்றும் அதனுடைய பதினாறடி எண்ணும் கொடுக்கப்பட்டுள்ளது. திட்டநிரலிலிருந்து ஒரு குறிப்பிட்ட அழைப்பிடத்திற்கு மாற்ற **INTR** புற வன்மியம் பயன்படுகிறது. இந்த நோக்கத்திற்காக வன்மியம் சுற்று RST குறியீட்டை உண்டாக்குகிறது. **INTR** உயர்நிலையில் இருக்கும்போது, நுண்முறைப்படுத்தியானது திட்டநிரல் எண்ணியில்

உள்ளவற்றை அடுக்கில் சேமிக்கிறது. **INTR**-இன் பதிலுக்கு ஏற்றவாறு புற வன்மியம் **RST n** குறியீட்டை உண்டாக்குகிறது. இந்தக் குறியீட்டை நுண்முறைப்படுத்தி பெற்றவுடன், அது திட்டநிரலைப் பொருத்தமான அழைப்பு இடத்திற்கு மாற்றுகிறது. எட்டு **I/O** கருவிகள் வரைக்கும் புற வன்மியத்தின் வழியாக **INTR** உடன் இணைக்கலாம். **INTR** உடன் இணைக்கப்பட்டுள்ள **I/O** கருவிகளுக்குப் புற வன்மியம் அல்லது குறுக்கீடு கட்டுப்பாட்டாளர் வழியாக முந்துரிமை ஒப்படைக்கப்படுகிறது. புற வன்மியமானது, எந்த **I/O** கருவி குறுக்கிட்டது என்பதை அறிந்துகொள்கிறது. அது சரியான குறியீட்டை உண்டாக்குவதால், நுண்முறைப்படுத்தி ஒரு குறிப்பிட்ட **I/O** கருவியை எடுப்பதற்குக் காரணமாகிறது. அதற்குப் பதிலாக முந்துரிமைக் குறுக்கீடு கட்டுப்பாட்டாளர் **8214** ஐ பயன்படுத்த உதவுகிறது. திட்டமிடத்தக்க குறுக்கீடு கட்டுப்பாட்டாளர் **8259** ஐ அதிக சிறப்பானது என்பதோடு, அதுவே உபயோகத்திலும் உள்ளது. **8259** அல்லது மற்ற சில வன்மியத்தை ஊக்குவிப்பதற்கு **INTA** உபயோகப்படுகிறது.

2.1. 4. RST 7.5, 6.5 மற்றும் 5.5 :

RST 7.5, 6.5 மற்றும் **5.5** ஆகியன குறுக்கீடுகள். இந்தக் குறுக்கீடுகள் மென்மியத்தில் உள்ள கட்டளைகள் **EI** மற்றும் **SIM (Set Interrupt Mask)** மூலமாகச் செயலியக்கம் பெறுகின்றன. சேமக்கலத்தில் கட்டங்கள் அமைப்பிற்கு ஏற்றவாறு **SIM** கட்டளை இயக்கத்தின்போது குறுக்கீடுகளின் செயலியக்கமோ/செயல் முடக்கமோ அமைகின்றன. படம் **2.2**இல் **SIM** கட்டளைக்கான சேமக்கலத்தின் பொருளடக்கம் காட்டப்பட்டுள்ளது.



படம் **2.2 SIM**- க்கான சேமக்கலத்தின் உள்ளடக்கம்

முனை 0 RST 5.5க்காகவும், முனை 1 RST 6.5க்காகவும் மற்றும் முனை 2 RST 7.5க்காகவும் உள்ளது. பிட் 1 என அமைந்தால், ஒத்திருக்கிற குறுக்கீடு செயல்முடக்கம் அடைகிறது. 0 எனில் ஒத்திருக்கிற குறுக்கீடு செயலியக்கம் பெறுகிறது. முனை 0–2ஐ சக்தி வாய்ந்ததாக மாற்ற முனை 3, 1 ஆக அமைய வேண்டும். 4 RST 7.5இன் கூடுதல் அதிகார முனையாகும். இது 1 எனில், RST 7.5கான நிலைமாறி மறுஅமைவு பெறும். RST 7.5க்கான முனை 0 அல்லது 1 என எதுவாக இருந்தாலும் RST 7.5 செயல் முடக்கம் அடைகிறது.

அட்டவணை 2.1 RST n க்கான அழைப்பு இடங்கள் மற்றும் பதினாறடி குறியீடுகள்.

RST n	பதினாறடி குறியீடு	அழைப்பு இடங்கள்
RST 0	C 7	0000
RST 1	C F	0008
RST 2	D 7	0010
RST 3	D F	0018
RST 4	E 7	0020
RST 5	E F	0028
RST 6	F 7	0030
RST 7	F F	0038

6 மற்றும் 7 முனைகள் தொடர் செய்தித் தொகுப்பு என்பதைக் குறிக்கின்றன. கட்டளை SIM-உம் தொடர் செய்தியை அனுப்புவதற்குப் பயன்படுத்தப்படுகிறது. முனை 6 SODஐ செயலியக்கம் செய்கிறது. கட்டளை SIMஐ செய்து முடிக்கும்போது சேமக்கலத்தின் 7ஆவது முனை வெளியீடாக நுண்முறைப்படுத்தியிலுள்ள SOD வரியில் கிடைக்கும். 7ஆவது முனை உயர்நிலை (1) அல்லது (0) ஆகவும் இருக்கலாம்.

கட்டளை DF அனைத்துக் குறுக்கீடுகளையும் செயல் முடக்கம் செய்கிறது. எப்பொழுதும் இது விரும்பத்தக்கதல்ல. நுண்முறைப்படுத்தி ஒரு குறிப்பிட்ட வேலையைச் செய்து கொண்டிருக்கும்போது, சில குறுக்கீடுகளின் நிகழ்வினைத் தடுப்பதே விரும்பத் தகுந்தனவாகும். அந்தச் சமயங்களில்

விரும்பத்தகாத குறுக்கீடுகளின் நிகழ்வினை SIM கட்டளையின் மூலமாகச் செயல்முடக்கம் செய்யவேண்டும்.

தொடக்கி மட்டங்கள் :

குறுக்கீடுகள் விளிம்புத் தொடக்கி, மட்டத் தொடக்கி அல்லது விளிம்பு மற்றும் மட்டம் கொண்ட தொடக்கி என்கிற சைகைகளினால் தேர்வு செய்யப்படுகின்றன. விளிம்பு மற்றும் மட்டமும் கொண்ட சைகையின் மூலம் TRAP குறுக்கீடு செய்யப்படுகிறது. இது என்னவென்றால், TRAP குறுக்கீடு உயர்நிலைக்குச் சென்று ஒப்புதல் வரும்வரை உயர்நிலையிலேயே இருக்கும். இதனால் இரைச்சல் மற்றும் நிலையற்ற நிலை போன்றவைகளினால் ஏற்படுகின்ற தவறுதலான தொடக்கி தவிர்க்கப்படுகிறது. நேர் விளிம்பு கொண்ட ஒரு சைகையின்மூலம் RST 7.5 தொட க்கப்படுகிறது. இது சிறிய நீடிக்கும் காலம் உடைய சைகையின் மூலம் தொடங்குகிறது. RST 6.5 மற்றும் RST 5.5 மட்டச் சைகையின் மூலம் தொடக்கப்படும். RST 6.5 மற்றும் RST 5.5 குறுக்கீடுகள் உயர்மட்டச் சைகையின் மூலம் தொடங்கி வைக்கப்படுகின்றன. இதில், நுண்முறைப்படுத்தியானது குறுக்கீடை ஏற்றுக்கொள்கின்றவரை சைகை உயர்நிலையில் அப்படியே இருக்கும்.

உதாரணம் 1 : 8085இலுள்ள எல்லா குறுக்கீடுகளையும் செயலியக்கம் செய்ய வேண்டும். RST 7.5. RST 6.5 மற்றும் RST 5.5 குறுக்கீடுகள் கட்டளைகள் SIM மூலம் செயலியக்கம் பெறுவதற்குச் சேமக்கலத்தினுள் அடங்கியவற்றைக் கீழ்க்கண்டவாறு திட்டநிரல் செய்யவேண்டும்.

7	6	5	4	3	2	1	0
SOD	DOE	X	R 7.5	MSE	M 7.5	M 6.5	M 5.5
0	0	0	0	1	0	0	0 = 08

RST 7.5, RST 6.5 மற்றும் RST 5.5 செயலியக்கம் பெற பிட்டுகள் 0, 1 மற்றும் 2 பூஜ்ஜியமாக அமைய வேண்டும். பிட்டுகள் 0, 1, 2–ஐ சக்தி வாய்ந்தவைகளாக . மாற்றப் பிட் 3 அமைவு நிலையில் இருக்கவேண்டும். RST 7.5 செயலியக்கம் பெற பிட் 4–ஐ

பூஜ்ஜியமாக்க வேண்டும். ஆகையால் **RST 7.5** கூடுதல் அதிகாரம் பெறுகிறது.

கட்டளைகள்

செயற்குறிப்பு	இயமிகள்	விளக்கம்
EI		எல்லா குறுக்கீடுகளும் செயலியக்கம் பெறுதல்.
MVI	**A, 08**	**RST 7.5, 6.5** மற்றும் **5.5** செயலியக்கம் அடையச் சேமக்கலம் பிட் முறையைப் பெற
SIM		**RST 7.5, 6.5** மற்றும் **5.5** களின் செயலியக்கம்

கட்டளைகள் **IE** மற்றும் **SIM** இரண்டும், **RST 7.5, 6.5** மற்றும் **5.5** குறுக்கீடுகள் செயலியக்கம் அடைய முக்கியமானவை. இவையல்லாமல், கூடுதலாக **INTR** ஐ கட்டளை **EI** செயலியக்கம் அடையச் செய்கிறது.

உதாரணம் 2 : RST 6.5 ன் செயலியக்கம் மற்றும் **RST 7.5** மற்றும் **5.5** இன் செயல்முடக்கம்.
சேமக்கலத்தினுள் அடங்கியவை:

7	6	5	4	3	2	1	0
SOD	SOD	X	R 7.5	MSE	M 7.5	M 6.5	M 5.5
0	0	0	1	1	1	0	1 = 1D

RST 6.5 செயலியக்கம் பெற பிட் 1 ஐ பூஜ்ஜியமாக்க வேண்டும். **RST** 5.5 மற்றும் 7.5 ஆகியவை முறையே செயல் முடக்கம் அடைய பிட் 0 மற்றும் 2 ஐ 1 ஆக மாற்றவேண்டும். **RST 7.5**க்குக் கூடுதல் அதிகாரம் கொண்ட பிட் 4ஐ, **RST 7.5** செயல் முடக்கம் பெற 1 ஆக்க வேண்டும். பிட் 0, 1 மற்றும் 2 சக்தி வாய்ந்ததாக உருவாக்க பிட் 3 ஐ 1 ஆக அமைக்க வேண்டும்.

கட்டளைகள் :

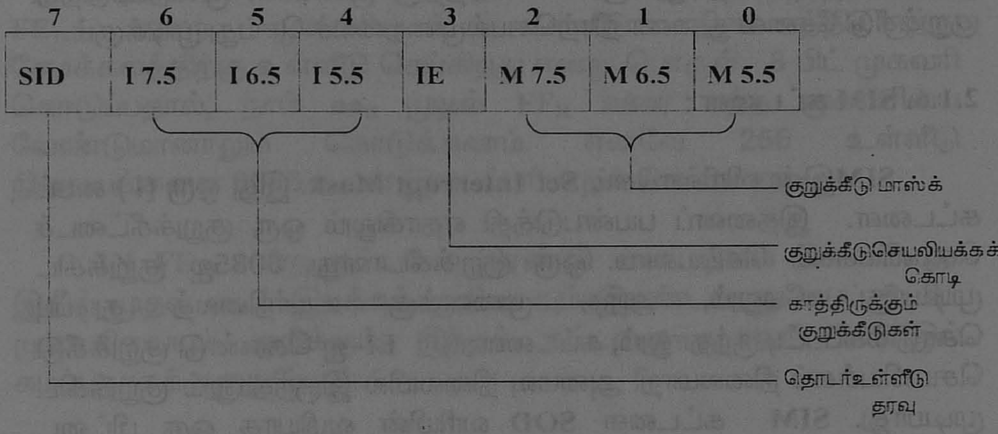
செயற்குறிப்பு	இயமிகள்	விளக்கம்
EI		குறுக்கீடுகள் செயலியக்கம் பெறுதல்.
MVI	A, 1D	RST 6.5 செயலியக்கம் பெறுவதற்கு RST7.5 மற்றும் RST 5.5 செயல் முடக்கம் அடைவதற்கான சேமக்கலம் முறையைப் பெற.
SIM செயலியக்கம்		RST 6.5 இன் மற்றும் 7.5 மற்றும் 5.5 களின் செயல் முடக்கம்.

முடிவு பெறாத குறுக்கீடுகள் :

ஒரு குறுக்கீடின் வேண்டுகோள் நிகழும்போது, மற்ற குறுக்கீடுகளின் நிகழ்வு முடிவு பெறாத வேண்டுகோளாகும். ஒரே சமயத்தில் ஒன்றுக்கும் மேற்பட்ட குறுக்கீடுகள் நேரிடும்போது, முதன்மை முந்துரிமைப் பெற்ற குறுக்கீடு செயல்படுத்தப்படுகிறது. மற்ற தாழ்ந்த முந்துரிமைக் குறுக்கீடுகள் முடிவு பெறாமல் தொடர்ந்து இருக்கின்றன. 8085 அதிக குறுக்கீடு வரிகளைக் கொண்டது. ஒரு குறுக்கீடின் சேவை நடந்துகொண்டிருக்கும்போது நேரிடுகின்ற மற்ற குறுக்கீடுகள் முடிவுபெறாத குறுக்கீடுகளாகும்.

2.1. 5. RIM கட்டளை :

8085 இல் உள்ள RIM கட்டளையின்மூலம், திட்ட அமைப்பாளர் தற்போதைய முடிவு பெறாத குறுக்கீடுகளின் அந்தஸ்தைப் பற்றித் தெரிந்து கொள்ளலாம். RIMஇன் விரிவாக்கம் **Read Interrupt Mask** (மறைக்கப்பட்ட குறுக்கீடு படி) ஆகும். இந்தக் கட்டளையின் மூலம் தற்போதைய குறுக்கீடுகளின் அந்தஸ்தைப் பற்றி அறியலாம்.



படம் **2.3** RIM கட்டளை செய்து முடித்ததற்குப் பின் சேமக்கலத்தின் உள்ளடக்கம்

INTRஇன் நிகழ்ச்சியின்போது, முறைப்படுத்தித் தன்நிலையை ஒரு கட்டளையின் கடைசி எந்திரச் சுழற்சியின் கடைசி நிலையிலிருந்து படிக்கிறது. முறைப்படுத்தித் தலைமைத் திட்டநிரலுக்குத் திரும்பும்போது, **INTR**இன் வரிநிலை தலைமைத் திட்டநிரலை மீண்டும் செய்வதற்கு முன் தெரிந்து கொள்ளும்.

படம் 2.3 இல் சேமக்கலத்தின் உள்ளடக்கம் கட்டளை செய்து முடித்தற்குப் பின் காட்டப்பட்டுள்ளது. இந்தக் கட்டளை 8 பிட் சேமக்கலத்தை நிரப்புகிறது. இந்த 8 பிட்களைப் பயன்படுத்தி, குறுக்கீடு மறைப்பின் **(Mask)** அந்தஸ்தைக் காணலாம். **RIM** கட்டளையைப் பயன்படுத்தி முடிவுபெறா குறுக்கீடுகளை அடையாளம் காண முடியும். இது தொடர் செய்தியை ஏற்றுக்கொள்வதற்கும் பயன்படுகிறது. பிட்டுகள் 0 – 2 குறுக்கீடு மறைப்பு; **1** = மறைக்கப்பட்டது. பிட் 3 குறுக்கீடு செயலியக்கக் கொடியைக் குறிப்பிடுகிறது; 1 = செயலியக்கம். பிட்டுகள் 4 – 6 முடிவு பெறாத குறுக்கீடுகளைச் சுட்டிக்காட்டுகிறது; 1 = முடிவு பெறாதது. பிட் 7 ஆனது தொடர் உள்ளீடு தரவிருந்தால் அதனைக் குறிக்கிறது.

குறுக்கீடு சேவை துணை நிரற்செயல்முறையைச் செய்து முடித்த பின், முறைப்படுத்திக் குறுக்கீடுகள் ஏதாகிலும் முடிவு பெறாமல் இருக்கின்றனவா என்று **RIM** கட்டளையைப் பயன்படுத்தித் தணிக்கை செய்கிறது. குறுக்கீடுகள் முடிவு பெறாமலிருந்தால்,

துலைமைத் திட்டநிரலுக்குச் செல்வதற்கு முன்பாக, முறைப்படுத்தி குறுக்கீடு சேவை துணை நிரற்செயல்முறையைச் செய்து முடிக்கும்.

2.1.6. SIM கட்டளை :

SIMஇன் விரிவாக்கம் **Set Interrupt Mask.** இது ஒரு (1) பைட் கட்டளை. இதனைப் பயன்படுத்தி ஏதாகிலும் ஒரு குறுக்கீட்டைச் செயலியக்கம் செய்யலாம். ஒரு குறுக்கீடானது, 8085ஐ குறுக்கிட முடியாது. மேலும், அந்த முனைக்கு உயர்நிலைத் துடிப்பு கொடுக்கப்பட்டிருந்தாலும், கட்டளை **EI**–ஐ கொண்டு குறுக்கீடு செயலியக்கம் நிலைமாறி அமைவு நிலையில் இருந்தாலும் குறுக்கிட முடியாது. **SIM** கட்டளை **SOD** வரியின் வழியாக ஒரு பிட்டை அனுப்புகிறது.

2.2. I/O இடைப்பரிமாறல் :

விசைப்பலகை உள்ளீடு நிலையத்தின் வழியாகவும், **CRT** மானிட்டர் வெளியீடு நிலையத்தின் வழியாகவும் இணைக்கப்படலாம். 8085லுள்ள **I/O** இடைப்பரிமாறல் வெளி உலகத்துடன் தொடர்புகொள்ள அனுமதிக்கிறது. இரண்டு வகையான **I/O** இடைப்பரிமாறல்கள் உள்ளன. அவையாவன,

- நேர் **I/O or** எளிய **I/O or** வரையப்பட்ட **or** புறவியலான **I/O**
- நினைவகம் வரையப்பட்ட **I/O**

2.2.1 உள்ளீடு கட்டளை மற்றும் அதன் நேரமுறை வரைபடம் :

உள்ளீடு செய்தியை உள்ளீடு நிலையத்திலிருந்து சேமக்கலத்திடம் உள்ளீடு செய்வதற்கு 8085–ல் பயன்படுத்தும் கட்டளை.

IN Addr 8

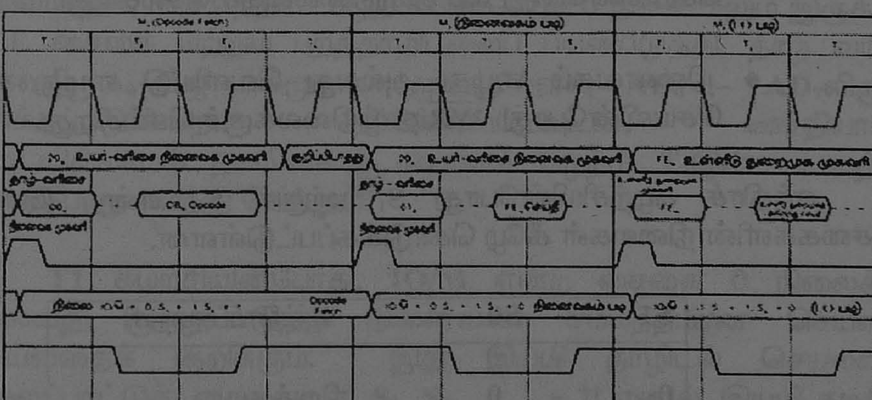
இது ஒரு பைட்டை உள்ளீடு நிலையத்திலிருந்து சேமக்கலத்திடம் மாற்றுகிறது. உள்ளீடு நிலையத்தின் முகவரி கட்டளையிலேயே உள்ளது. **I/O** கட்டளைகளின் நிலைய முகவரி **8** பிட்டுகள் கொண்டது.

உதாரணமாக, **IN FE** என்ற கட்டளையில் 8 பிட் தரவினை FE_H என்ற முகவரியுடன் உள்ளீடு நிலையத்திலிருந்து சேமக்கலத்திற்கு உள்ளீடு செய்கிறது என்று பொருள். 8 பிட் முகவரி கொடுப்பதால், நாம் 00_H முதல் FF_H வரை எந்த முகவரியை வேண்டுமானாலும் கொடுக்கலாம். எனவே 256 உள்ளீடு நிலையங்களை 8085 உடன் இடைப்பரிமாறல் செய்யலாம்.

IN FE_{16} என்ற கட்டளை இரண்டு பைட் நீளம் கொண்டது. இதில் முதல் பைட் இயக்கக் குறியையும் இரண்டாவது பைட் 8 பிட் முகவரியையும் குறிக்கும். இந்தக் கட்டளை முகவரி 2000இல் சுமப்பதாகக் கருதுவோம்.

2000	**DB**
2001	**FE**

8085இன் மூலம் எவ்வாறு இந்தக் கட்டளை செய்யப்படுகிறது என்று படிப்படியாகப் பார்ப்போம். **IN** கட்டளைக்கான கட்டளைச் சுழற்சி மூன்று எந்திரச் சுழற்சியைக் கொண்டது. அது **10 T** - நிலைகள் எடுத்துக் கொள்கிறது. **IN** FE_H என்ற கட்டளைக்கான நேரமுறை வரைபடம் படம் 2.4இல் காட்டப்பட்டுள்ளது.



படம் 2.4 **IN** FE_H கட்டளைக்கான நேரமுறை வரைபடம்

- A_8 முதல் A_{15} வரையுள்ள உயர்நிலை வரிசை முகவரியைத் தனிப்பட்ட எட்டு வரிகளில் 8085 அனுப்புகிறது.

- தாழ்நிலை வரிசை முகவரி $A_0 - A_7$ மற்றும் தரவினை $D_0 - D_7$ பல்வின் ஒன்றாக்கி $AD_0 - AD_7$ ஆக அனுப்பப்படுகிறது.

- 8085 ALE என்ற சைகையை ஒவ்வொரு எந்திரச் சுழற்சியின் தொடக்கத்தின்போதும் அனுப்புகிறது. இது புறத்தாழிடலுடன் சேர்ந்து, தாழ்நிலை வரிசை முகவரியையும் தரவினையும். பிரிப்பதற்குப் பயன்படுகிறது.

- நுண்முறைப்படுத்தி நினைவகம் சம்மந்தப்பட்ட செயலைச் செய்யும்போது $IO/\overline{M}$ சைகை 0 நிலைக்குச் செல்கிறது. உள்ளீடு / வெளியீடு சம்மந்தப்பட்ட செயலைச் செய்யும்போது $IO/\overline{M}$ சைகை 1 நிலைக்குச் செல்கிறது.

- நினைவகம் படி அல்லது உள்ளீடு படி என்ற செயலின்போது $\overline{RD}$ தாழ்நிலைக்குச் செல்கிறது.

- நினைவகம் எழுது அல்லது வெளியீடு எழுது என்ற செயலின்போது $\overline{WR}$ தாழ்நிலைக்குச் செல்கிறது.

எந்திரச் சுழற்சியின்போது S_1 மற்றும் S_O என்ற இரண்டு சைகைகளின் நிலைகள் கீழே கொடுக்கப்பட்டுள்ளன.

S1	S0	எந்திரச் சுழற்சி
0	0	நிறுத்து
0	1	எழுது
1	0	படி
1	1	இயமி கொணர்

IN FE_H கட்டளைக்கான கட்டளைச் சுழற்சி மூன்று எந்திரச் சுழற்சிகளைக் கொண்டது. அவையாவன,

1. இயமி கொணர்
2. நினைவகம் படி
3. I/O படி

1. <u>இயமி கொணர் எந்திரச் சுழற்சி</u> :

இயமி கொணர் எந்திரச் சுழற்சி நான்கு கடிகாரச் சுழற்சியில் அல்லது T_1, T_2, T_3 மற்றும் T_4 என்ற நான்கு T நிலைகளில் முடிந்துவிடும். இயமி தாழிடல் சுழற்சியின்போது, முறைப்படுத்திக் கட்டளையின் இயமியை அது சேமித்து வைக்கப்பட்டிருந்த நினைவக இடத்திலிருந்து கொண்டு வருகிறது.

முதலாவது கடிகாரச் சுழற்சி T_1 ன் போது, முறைப்படுத்தி உயர்நிலை வரிசை முகவரி (20_H) யை A_8 – A_{15} வரிசைகளில் வெளியிடும். இந்த முகவரியானது மீதமுள்ள எந்திரச் சுழற்சி வரைக்கும் மாறாமல் அப்படியேயிருக்கும். அதே சமயத்தில், தாழ்நிலை வரிசை முகவரி (00_H)யை பல்முனை முகவரி/தரவு வரிகள் AD_0 – AD_7 இல் அனுப்புகிறது. எந்திரச் சுழற்சியின்போது 8085, ALE என்ற சைகையை T_1இன் போது அனுப்பும். ALE பாதி நேரத்தில் 1 நிலையிலும், மீதி எந்திரச் சுழற்சியின்போது 0 நிலையிலும் இருக்கும். ALE சைகை மற்றும் புறத்தாழிடலைப் பயன்படுத்தி, தாழ் வரிசை முகவரி A_0-A_7ஐ பல்முனை முகவரி/தரவு AD_0 – AD_7லிருந்து பிரிக்கப்படுகின்றன. பிரிக்கப்பட்டவுடன், ஊர்தியானது தடையில்லாமல் தரவினைக் கடிகாரச் சுழற்சி T_2இன் மையத்தில் வரும்போது சுமந்து செல்லும்.

T1 சுழற்சியின்போது, $IO/\overline{M}$ என்ற சைகை 0 நிலைக்குச் செல்வது முறைப்படுத்தி நினைவகச் சம்மந்தமான செயலைச் செய்வதைக் குறிக்கும். இது இயமி தாழிடல் செயலாகும். நிலைப்பாட்டுச் சைகைகள் S_1 S_0 = 11 எனில், இயமி தாழிடல் எந்திரச் சுழற்சி என்பதைக் குறிக்கும். $IO/\overline{M}$ மற்றும் S_1 S_0 சைகைகள் மீதமுள்ள நேரம் வரைக்கும் அதே நிலையிலேயேயிருக்கும்.

T_1 சுழற்சியின் $\overline{RD}$ சைகை முதலில் உயர்நிலையிலும் (1) T_2 சுற்றின் மையத்தில் வரும்போது தாழ்நிலை(0)க்கும் செல்கிறது. T_3 சுற்று முடியும் வரையிலும் $\overline{RD}$ சைகை தாழ்நிலையிலேயேயிருக்கும். $IO/\overline{M} = 0$ மற்றும் $\overline{RD} = 0$ எனில் இந்த சைகைகள் தனியாகவோ அல்லது சேர்ந்தோ $\overline{MEMR}$ சைகையைப் பெற்று நினைவகத்திலிருந்து படிப்பதற்குப் பயன்படுகிறது.

T_2 சுழற்சியின் மையத்தில், நினைவக முகவரி 16 பிட் முகவரி ஊர்தியில் கிடைக்கும் மற்றும் நினைவக எழுது சைகையும் தயாராக இருக்கும். நினைவகத்தில் உள்ளவை படிக்கப்படுகின்றன. அவ்வமயம் தரவு ஊர்தி D_0 - D_7 இல் தரவானது வீழ்கின்றது. இப்போது தரவு ஊர்தியானது நினைவக இருப்பிடத்திலிருந்து (2000_H) படித்து இயமியைப் (DB) பெறும். கட்டளையின் இயமியானது 8085–இல் உள்ள கட்டளைப் பதிவேட்டைச் சென்றடைகின்றது. T_4 சுழற்சியில் குறியீடு பெயர்ப்பு அங்கு நடைபெறுகிறது.

2. நினைவகம் படி எந்திரச் சுழற்சி:

மூன்று கடிகாரச் சுழற்சிகளில் நினைவக எந்திரச் சுழற்சி பூர்த்தி அடைகிறது. பினைவகம் படி சுழற்சியில், நுண்முறைப்படுத்தி தரவினை நினைவக இருப்பிடத்திலிருந்து படிக்கின்றது. திட்டநிரல் எண்ணியிலுள்ள முகவரி தானாகவே உயர்ந்து 2001_H என்ற முகவரியை அடையும்.

முதல் கடிகாரச் சுழற்சி T_1இன் போது, முறைப்படுத்தி உயர்நிலை வரிசை முகவரி (20_H)ஐ $A_8 - A_{15}$ வரிகளில் வெளியீடு செய்கிறது. இந்த முகவரி எந்திரச் சுற்று முழுவதும் மாறாமல் இருக்கும். அதே சமயத்தில், தாழ் வரிசை முகவரி (01_H) பல்முனை முகவரி/தரவு வரிகளின் மூலமாக அனுப்பப்படுகிறது. T1 சுழற்சியின்போது 8085 முறைப்படுத்தி ALE சைகையை அனுப்பும்.

T2 சுற்றின் மையத்தில், நினைவக முகவரி 16 பிட் முகவரி ஊர்தியில் கிடைக்கும். மற்றும் நினைவக எழுது சைகையும் தயாராய் இருக்கும். நினைவகத்தில் உள்ளவை படிக்கப்படுகின்றன. தரவுத் தொகுப்பானது தரவுத் தொகுப்பு ஊர்தியில் D_0 - D_7 இல் வீழ்கின்றன. இப்போது தரவுத் தொகுப்பு ஊர்தியானது தரவுத் தொகுப்பை

நினைவக இருப்பிடம் 2001_H லிருந்து பெறும். தரவுத் தொகுப்பானது உள்ளீடு நிலையத்தின் 8 பிட் முகவரியாகும். அடுத்த எந்திரச் சுழற்சி பயன்பாட்டிற்காக இந்த நிலையத்தின் முகவரி 8085இன் உள்ளே தற்காலிகமாகச் சேமிக்கப்படும்.

3. I/O படி எந்திரச் சுழற்சி :

I/O படி எந்திரச் சுழற்சிக்கு மூன்று கடிகாரச் சுழற்சிகள் தேவை. இந்த I/O எழுது சுழற்சியில் முறைப்படுத்தித் தரவை உள்ளீடு நிலையத்திலிருந்து படிக்கும். I/O நிலையத்தின் 8 பிட் முகவரியை உயர் வரிசை முகவரி வரிகள் A_8-A_{15} இன் மீதும் மற்றும் தாழ் வரிசை முகவரி வரிகள் A_0 - A_7 இன் மீது அனுப்பும்.

முதல் கடிகாரச் சுழற்சி T_1இன் போது, முறைப்படுத்தி உள்ளீடு நிலையத்திலிருந்து முகவரியை (FE_H) A_8 – A_{15} வரிகளின்மூலம் வெளியீடு செய்யும். எந்திரச் சுழற்சி முழுவதும் இந்த முகவரி மாறாமல் அப்படியே இருக்கும். அதே முகவரி (FE_H) பல்முனை முகவரி/தரவு வரிகள் AD_0 – AD_7 இன் தாழ் வரிசை முகவரி A_0 – A_7 உடன் செல்லும்.

T_1இன்போது $\overline{RD}$ சைகை முதலில் உயர்நிலையிலும் T_2இன் மையத்தில் தாழ்நிலைக்கும் செல்லும். T_3 முடியும் வரை $\overline{RD}$ சைகை தாழ்நிலையிலே இருக்கும். $IO/\overline{M} = 1$ மற்றும் $\overline{RD} = 0$ வாக இருக்கும்போது, இந்தச் சைகைகள் தனியாகவோ அல்லது சேர்ந்தோ $\overline{IOR}$ சைகை பெற்று, நினைவகத்திலிருந்து படிப்பதற்குப் பயன்படுகிறது.

T_2இன் மையத்தில் 8 பிட் முகவரி மற்றும் I/O படி சைகையுடன் முறைப்படுத்தித் தயாராய் இருக்கும். இந்தச் சமயத்தில், உள்ளீடு நிலையத்தில் உள்ளவற்றைத் தரவு ஊர்தியில் D_0 – D_7 வைப்பதற்கு ஏற்றாற்போல் உள்ளீடு நிலையப் பரிமாற்றுச் சுற்று திட்டமிடப்பட்டிருக்கும் உள்ளீடு நிலையத்திலிருந்து படிக்கப்பட்ட தரவானது தற்சமயம் சேமக்கலத்தைச் சென்றடையும். இப்போது, IN FE_H கட்டளைச் சுழற்சி முடிவடையும்.

2.2.2 வெளியீடு கட்டளை மற்றும் அதன் நேரமுறை வரைபடம் :

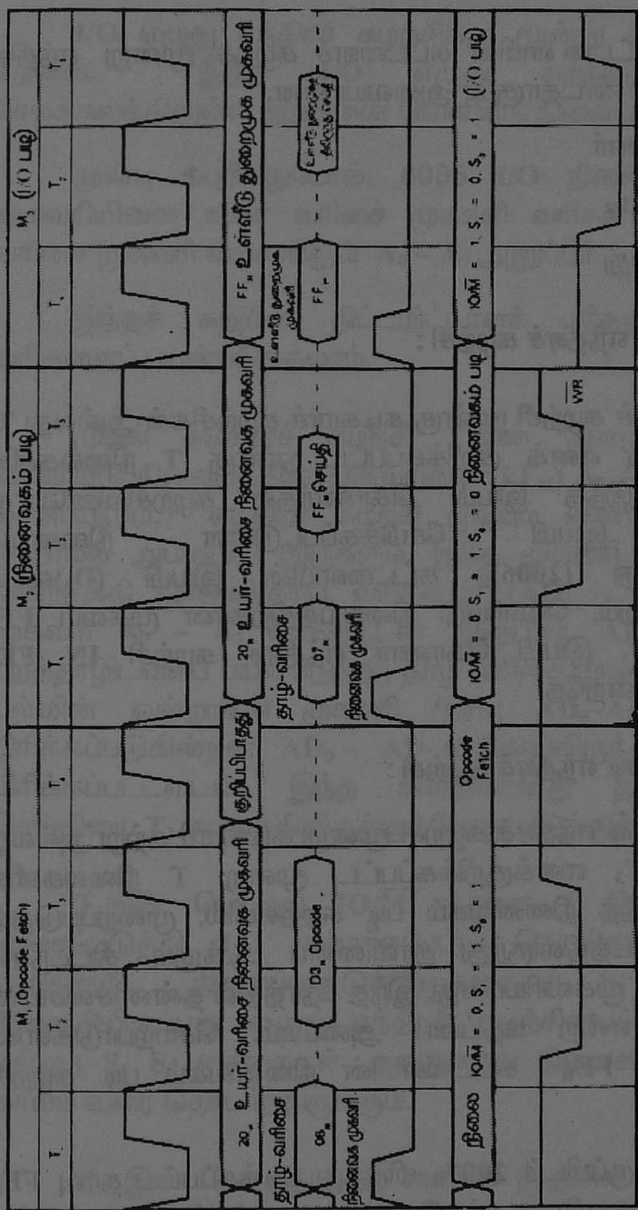
சேமக்கலத்திலிருந்து தரவினை வெளியீடு நிலையத்தில் வெளியிடுவதற்கு 8085 **OUT Addr 8** என்ற கட்டளையைப் பயன்படுத்தி ஒரு பைட்டைச் சேமக்கலத்திலிருந்து வெளியீடு நிலையத்திற்கு மாற்றுகிறது. வெளியீடு நிலையத்தின் முகவரி அந்தக் கட்டளையிலேயே கொடுக்கப்பட்டுள்ளது. **I/O** கட்டளைக்கான நிலைய முகவரி **8** பிட் கொண்டது.

உதாரணமாக, **OUT** FF_H என்ற கட்டளையில், 8 பிட் தரவைச் சேமக்கலத்திலிருந்து வெளியீடு நிலையத்தில் வெளியிடும். அதனுடைய முகவரி FF_H ஆகும். 8–பிட் முகவரி கொடுக்கப்பட்டுள்ளதால், நாம் 00_H முதல் FF_H வரை எந்த முகவரியை வேண்டுமானாலும் கொடுக்கலாம்.

OUT FF_H என்பது ஒரு இரு பைட் கட்டளை ஆகும். இதில் முதல் பைட் இயமியையும் **(D3)** இரண்டாவது பைட் **8** பிட் முகவரி FF_Hஐயும் குறிக்கும். இந்தக் கட்டளை 2006_H என்ற முகவரியில் சுமப்பதாகக் கருதுவோம்.

2006	**D3**
2007	**FF**

8085இன் மூலம் இந்தக் கட்டளை எவ்வாறு செயல்படுகிறது என்பதைப் படிப்படியாகப் பார்ப்போம். வெளியீடு கட்டளைக்கான கட்டளைச் சுழற்சி மூன்று எந்திர சுழற்சிகளையும் 10T நிலைகளையும் உள்ளடக்கியதாகும். **OUT** FF_H கட்டளையின் நேரமுறை வரைபடம் படம் 2.5இல் காட்டப்பட்டுள்ளது.



படம் **2.5 OUT FF$_H$** கட்டளைக்கான நேரமுறை வரைபடம்

OUT FF_H கட்டளைக்கான நேரமுறை வரைபடம் :

OUT FF_H கட்டளையின் கட்டளைச் சுழற்சி, மூன்று எந்திரச் சுழற்சிகளைக் கொண்டதாகும். அவையாவன,

1. இயமி கொணர்
2. நினைவகம் படி
3. **I/O** இல் எழுது

1. இயமி கொணர் எந்திரச் சுழற்சி :

இயமி எந்திரச் சுழற்சி நான்கு கடிகாரச் சுழற்சிகள் அல்லது **T_1, T_2, T_3** மற்றும் **T_4** எனக் குறிக்கப்பட்ட நான்கு **T** நிலைகளில் முடிவடையும். இந்த இயமி கொணர்தல் சுழற்சியின்போது, முறைப்படுத்தி இயமி சேமிக்கப்பட்டுள்ள நினைவக இருப்பிடத்திலிருந்து **$(2006)_H$** கட்டளையின் இயமி **(D_3)**யைத் தாழிடுகிறது. மேலும், வெளியீடு நிலையத்திற்கான முகவரி **FF_H** மாறுதலைத் தவிர, இயமி கொணர் எந்திரச் சுழற்சி **IN FE_H** கட்டளையைப் போன்றதே.

2. நினைவகம் படி எந்திரச் சுழற்சி :

நினைவகம் படி எந்திரச் சுழற்சி மூன்று கடிகாரச் சுற்று அல்லது **T_1, T_2** மற்றும் **T_3** எனக்குறிக்கப்பட்ட மூன்று **T** நிலைகளில் முடிவடையும். இந்த நினைவகம் படி சுழற்சியில், முறைப்படுத்தி நினைவக இருப்பிடத்திலிருந்து தரவினைப் படிக்கும். திட்டநிரல் எண்ணியில் உள்ள முகவரியானது, இந்த நேரத்தில் தன்னிச்சையாக உயர்ந்து, **2007_H** என்ற மதிப்பை அடையும். செயற்பாடுகளின் வரிசையானது **IN FE_H** கட்டளையின் நினைவகம் படி சுழற்சி போன்றிருக்கும்.

நினைவக இருப்பிடம் **2007_H** லிருந்து படிக்கப்பட்ட தரவு **FF_H** **8085** உள்ளே தற்காலிகமாகச் சேமித்து வைக்கப்படும். அடுத்த எந்திரச் சுழற்சிக்குப் பின்னர் இது பயன்படும்.

3. I/O எழுது எந்திரச் சுழற்சி :

I/O எழுது எந்திரச் சுழற்சிக்கு மூன்று கடிகாரச் சுழற்சிகள் தேவை. இந்த I/O எழுது சுற்றில், முறைப்படுத்திச் சேமக்கலத்திலுள்ள தரவினை வெளியீடு நிலையத்தில் எழுதும்.

முன்பு கூறியதுபோல், 8085 I/O நிலையங்களின் 8 பிட் முகவரியினை உயர் வரிசை முகவரி வரிகளிலும் A_8 –A_{15} தாழ் வரிசை முகவரி வரிகளிலும் A_0 – A_7 அனுப்பும்.

இந்தச் சுற்றின் திட்டமிட்டாளர் ஏதேனும் ஒரு முகவரி வரிகளைப் பயன்படுத்தலாம்.

முதல் கடிகாரச் சுழற்சி T_1இன் பொழுது முறைப்படுத்தி வெளியீடு நிலையத்தின் முகவரியை (FF_H) A_8 – A_{15} வரிகளின் மீது வெளியிடும். எந்திரச் சுழற்சி முடியும் வரையும் இந்த முகவரி மாறாமல் அப்படியே இருக்கும். இதே முகவரி (FF_H)யை, பல்வின் ஒன்றாக்கி முகவரி/தரவு வரிகளின் (AD_0 – AD_7) தாழ் வரிசை முகவரி A_0 – A_7 வரிகளுடன் அனுப்பும். ALE சைகை மற்றும் புறத்தாழிடலைப் பயன்படுத்தி, தாழ் வரிசை முகவரி A_0 – A_7 வரிகள் பல்வின் ஒன்றாக்கி முகவரி /தரவு AD_0–AD_7 வரிகளிலிருந்து பிரிக்கப்படுகின்றன. AD_0 – AD_7 வ ரிகளிலிருந்து A_0 – A_7 வரிகள் பிரிக்கப்பட்டவுடன், இந்த ஊர்தியானது தடையின்றித் தரவு செய்தியை T_2 மையத்திலிருக்கும்போது எடுத்துச் செல்லும்.

T_1இன் பொழுது $IO/\overline{M}$ சைகை 1 நிலைக்குச் செல்வது முறைப்படுத்தி I/O சம்மந்தப்பட்ட செயல்களைச் செய்வதைக் குறிக்கும். நிலைப்பாட்டுச் சைகைகள் S_1 S_0 முறையே 01 நிலையிலிருப்பது எழுது எந்திரச் சுழற்சியைக் குறிக்கும். $IO/\overline{M}$ மற்றும் S_1 S_0 சைகைகள் அதனுடைய நிலைகளிலேயே மீதமுள்ள காலம் வரை தொடர்ந்திருக்கும்.

T_1 இன் பொழுது $\overline{WR}$ சைகை முதலில் உயர்நிலைக்கும் (1) மற்றும் T_2 இ ன் மையத்தில் தாழ்நிலைக்கும் (0) செல்லும். T_3இன் முடிவு வரைக்கும் $\overline{WR}$ சைகை தாழ்நிலையிலேயே இருக்கும். $IO/\overline{M}$ = 1 மற்றும் $\overline{WR}$=0 எனில், இந்தச் சைகைகள் தனியாகவோ அல்லது

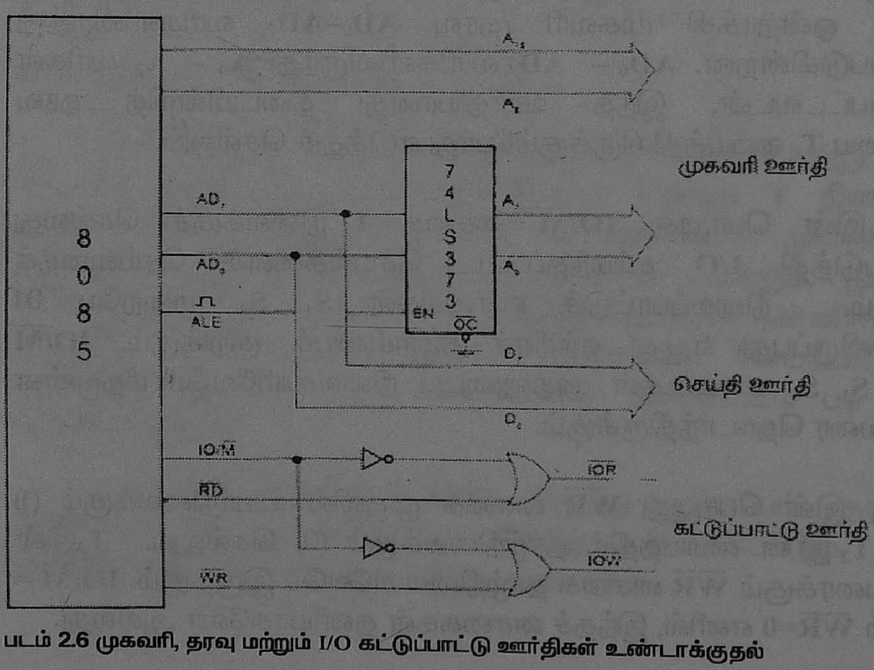
சேர்ந்தோ $\overline{IOW}$ சைகையைப் பெறுகிறது. இது வெளியீடு நிலையத்தில் எழுதுவதற்குப் பயன்படும்.

T_2 இன் மையத்தில், 8 பிட் முகவரி மற்றும் I/O எழுது சைகைகளுடன் 8085 தயாராய் இருக்கும். இப்பொழுது சேமக்கலத்தில் உள்ளவை தரவு ஊர்தியின் $D_0 - D_7$ மேல் விழுகிறது. OUT FF_H கட்டளைச் சுழற்சி நிறைவு பெறும்.

2.3 உள்ளீடு நிலையத் திட்டம் : (Direct I/O)

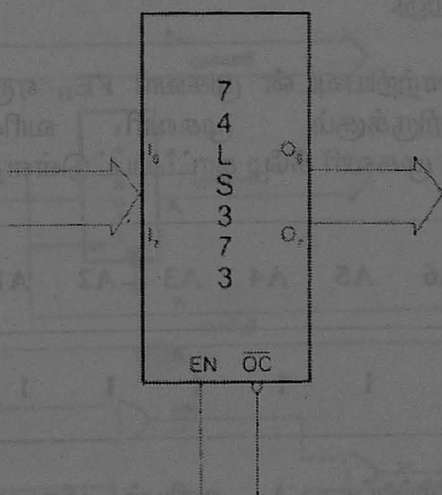
முதலில் தாழ் வரிசை முகவரி $A_0 - A_7$ மற்றும் தரவு $D_0 - D_7$ வரிகள் இவற்றைப் பல்வின் ஒன்றாக்கி முகவரி/தரவு வரிகள் $AD_0 - AD_7$ லிருந்து ஒன்றின் பலவாக மாற்ற வேண்டும். எட்டடி தாழிடல் (முகவரி தாழிடல்) மற்றும் ALE சைகையினைப் பயன்படுத்திச் செய்யலாம்.

I/O இடைப்பரிமாறலில், நாம் தனித்தனியாகச் சைகைகள், முறையே $\overline{IOR}$ சைகை I/O படிப்பதற்கும் $\overline{IOW}$ சைகை I/O எழுதுவதற்கும் உண்டாக்க முடியும். தனிப்படுத்தப்பட்ட முகவரி, தரவு ஊர்திகள் மற்றும் I/O கட்டுப்பாட்டுச் சைகைகள் படம் 2.6 இல் காட்டப்பட்டுள்ளது.



படம் 2.6 முகவரி, தரவு மற்றும் I/O கட்டுப்பாட்டு ஊர்திகள் உண்டாக்குதல்

முகவரி ஊர்தி, தரவு ஊர்தி மற்றும் கட்டுப்பாட்டு ஊர்திகள் உண்டாக்கியவுடன், அவற்றை உள்ளீடு நிலையமாகச் சேவை செய்யக்கூடிய ஓர் இலக்க வாதியல் கருவியுடன் சரியாக இணைக்க வேண்டும். அந்தக் கருவி எட்டு உள்ளீடுகளையும், எட்டு வெளியீடுகளையும் கொண்டதாக இருக்கும். கருவியின் வெளியீட்டை 8085இன் தரவு ஊர்தியுடன் இணைக்க வேண்டும். அவ்வாறு இணைப்பதால், அது இயற்கையாகவே உயர் நடைதடை வாய்ப்பைப் பெறும். கருவியானது சிலவிதமான வெளியீட்டுக் கட்டுப்பாட்டைப் பெற்றிருக்க வேண்டும். அதனால், எப்பொழுதாவது வெளியீட்டுக் கட்டுப்பாடு செயலியக்கம் அடையும்போது, கருவியானது தன் உள்ளீடை வெளியீடு வரிகள் மற்றும் தரவு ஊர்தியில் மாற்றும். கருவியின் வெளியீடு கட்டுப்பாட்டுச் செயலியக்கம் பெறாத பொழுது, வெளியீட்டு வரிகள் உயர் நடைதடை நிலைக்கு செல்லும். அப்போது, அது தன்னை, தரவு ஊர்தியிலிருந்து தனிமைப்படுத்திக்கொள்ளும். எட்டடி தாழிடல் உள்ளீடு நிலையமாகவும், வெளியீடு நிலையமாகவும் பயன்படுகிறது. எட்டடி தாழிடலின் கட்ட வரை படம் 2.7இல் காட்டப்பட்டுள்ளது.



படம் 2.7 எட்டடி தாழிடல்

EN மற்றும் $\overline{OC}$ என இரு கட்டுபாட்டுச் சைகைகள் உள்ளன. **EN=1** எனில், உள்ளீடு தாழிடப்படும். $\overline{OC}=0$ எனில், வெளியீடு செல்லக்கூடிய நியாயமான தரவாகும் மற்றும் $\overline{OC}=1$ எனில், வெளியீடு உயர் நடைதடை நிலையிலிருக்கும்.

எட்டடி தாழிடலை உள்ளீடு நிலையமாகவும் பயன்படுத்தலாம். உள்ளீடு வரிகள், உள்ளீடு தரவைத் தருவதற்குப் பயன்படும். எட்டடி தாழிடலின் வெளியீடு வரிகள் 8085 உள்ள தரவு ஊர்தியுடன் இணைக்கப்படும். உள்ளீடு செயலியக்கம் **EN** இந்த அமைப்பிற்குத் தேவையில்லை. ஆகையால், அதனை 1 நிலையிலேயே நிரந்தரமாக நிறுத்த வேண்டும்.

IN FF_H கட்டளையைச் செயல்படுத்தும்பொழுது, நம்முடைய உள்ளீடு நிலையம் எட்டடி தாழிடல் உள்ளீடு தரவை ஊர்தித் தரவில் வைக்கும். அதனால் 8085 சேமக்கலத்திலிருந்து தரவைத் திரட்டும். **IN FE_H** கட்டளையின் நேரமுறை வரைபடத்தில், மூன்றாவது எந்திரச் சுழற்சியின்போது, முறைப்படுத்தி உள்ளீடு நிலையத்தின் முகவரியை $A_0 - A_7$ வரிகளிலும் $A_8 - A_{15}$ வரிகளிலும் வைக்கிறது. T_1 இன் போது $IO/\overline{M}$ உயர்நிலைக்கும் (1) மற்றும் T_2இன் போது $\overline{RD}$ தாழ்நிலைக்கும் செல்லும். $IO/\overline{M}$ மற்றும் $\overline{RD}$ சேர்ந்து $\overline{IOR}$ சைகையை உண்டாக்குகின்றன. FE_H மற்றும் $\overline{IOR}$ சைகையின் 8 பிட் முகவரி இணைந்து குறைவியக்கத் தேர்வுத் துடிப்புக் கருவியை உண்டாக்குகின்றன. இந்தச் சைகை எட்டடி தாழிடலின் $\overline{OC}$க்குக் கொடுக்கப்படுகிறது.

ஈரடியாக மாற்றியவுடன் முகவரி FE_H ஏழு 1களையும் ஒரு 0வையும் பெற்றிருக்கும். முகவரி வரிகளால் எடுத்துச் செல்லப்படுகின்ற முகவரி கீழே காட்டப்பட்டுள்ளது.

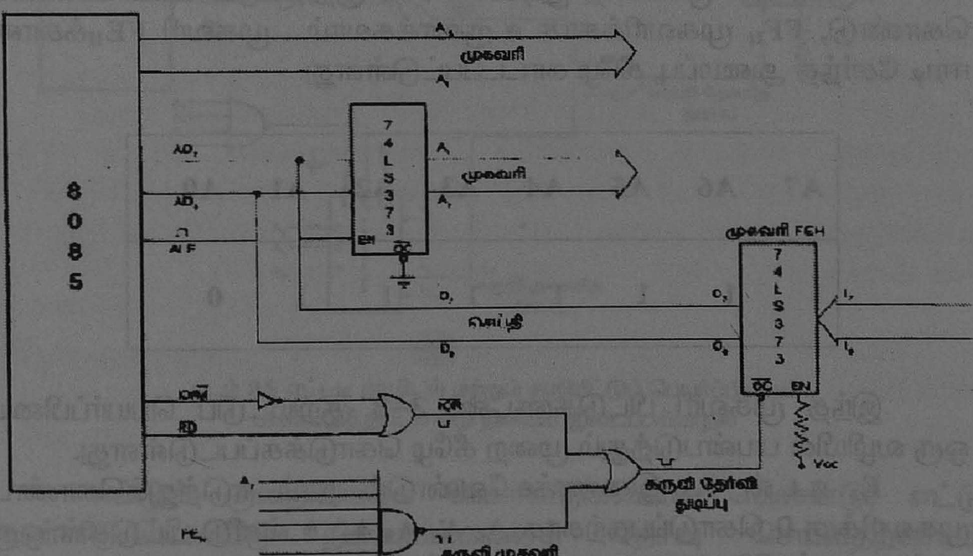
A7	A6	A5	A4	A3	A2	A1	A0
1	1	1	1	1	1	1	0

தலைகீழாக்கப்பட்ட A_0 வரியும் மீதமுள்ள ஏழு முகவரி வரிசைகளையும் அப்படியே எடுத்து 8 உள்ளீடு கொண்ட **NAND** கதவிற்கு உள்ளீடாகக் கொடுக்க வேண்டும். முகவரி FE_H க்கு மட்டும் **NAND** கதவின் வெளியீடு 0 நிலைக்குச் செல்லும். மற்ற வேறு முகவரிகளுக்கு **NAND** கதவின் வெளியீடு உயர்நிலையிலேயே (1)

இருக்கும். குறைவியக்கத் தாழ்ந்த துடிப்பினை உருவாக்குவதற்குக் கருவி முகவரித் துடிப்பு என்று பெயர்.

NAND கதவின் வெளியீடும் $\overline{IOR}$ சைகையும் **OR** கதவு **(7432)** வழியாக அனுப்பப்படும். **NAND** கதவின் வெளியீடு **0** மற்றும் $\overline{IOR} = 0$ எனில், **OR** கதவின் வெளியீடு **0** வாகும். **NAND** கதவின் வெளியீடு அல்லது $\overline{IOR}$ சைகை தாழ்நிலையிலில்லாதபோது அல்லது இரண்டுமே தாழ்நிலையிலில்லாதபோது, **OR** கதவின் வெளியீடு உயர்நிலையை (1) அடையும். FE_Hன் முகவரி மற்றும் $\overline{IOR} = 0$ எனில், கதவின் வெளியீடு 0 வாகும். உயர்நிலையிலிருந்து (1) தாழ்நிலைக்கு (0) மாறுவது என்பது கருவித் தேர்வுத் துடிப்பு ஆகும். இது உள்ளீடு நிலையம் செயலியக்கம் பெறுவதற்கு பயன்படுகிறது. அதனால், **OR** கதவின் வெளியீடு எட்டடி தாழிடலின் $\overline{OC}$ யுடன் சேர்க்கப்பட்டுள்ளது.

உள்ளீடு நிலைய இடைப்பரிமாறலின் முழுமையான சுற்றுப் படம் **2.8**இல் காட்டப்பட்டுள்ளது.



படம் 2.8 எட்டடி தாழிடலைப் பயன்படுத்தி உள்ளீடு நிலைய இடைப்பரிமாறல்

IN FE$_H$ என்ற கட்டளையைச் செயல்படுத்தும்போது, மூன்றாவது எந்திரச் சுழற்சியில் உள்ளீடு நிலையம் 8 பிட் தரவைத் தரவு ஊர்தியில் வைக்கும். பின் அது சேமக்கலத்திற்கு மாற்றப்படும்.

கருவியின் முகவரித் துடிப்பைப் பெறுவதற்கு ஒரு எட்டு உள்ளீடு **NAND** கதவு பயன்படுத்துவதற்குப் பதிலாக, ஒரு 3–8 குறியீட்டுப் பெயர்ப்பியைக்கூடப் பயன்படுத்தலாம். இந்தச் சில், எட்டுக் குறைவியக்க வெளியீடுகளைக் கொடுக்கும். இது எட்டு உள்ளீடு அல்லது வெளியீடு அல்லது இரண்டும் சேர்ந்த நிலையங்களை தேர்வு செய்வதற்குப் பயன்படும். 8 உள்ளீடு கொண்ட **NAND** கதவு பயன்படுத்த வேண்டிய இடத்தில், 3–8 குறியீட்டுப் பெயர்ப்பியை நாம் பயன்படுத்தலாம்.

3–8 குறியீட்டுப் பெயர்ப்பிக்காக, $\overline{E_1} = 0, \overline{E_2} = 0$ மற்றும் $E_3 = 1$ என நிர்ணயிக்க வேண்டும். பிறகு உள்ளீடு முகவரிக்கு A_3 A_2 A_1 **(C B A)** ஏற்றாற்போல், எட்டு வெளியீட்டில் ஏதேனும் ஒரு வெளியீடு 0 நிலைக்கும் மீதமுள்ள வெளியீடுகள் 1ஆம் நிலைக்கும் செல்லும்.

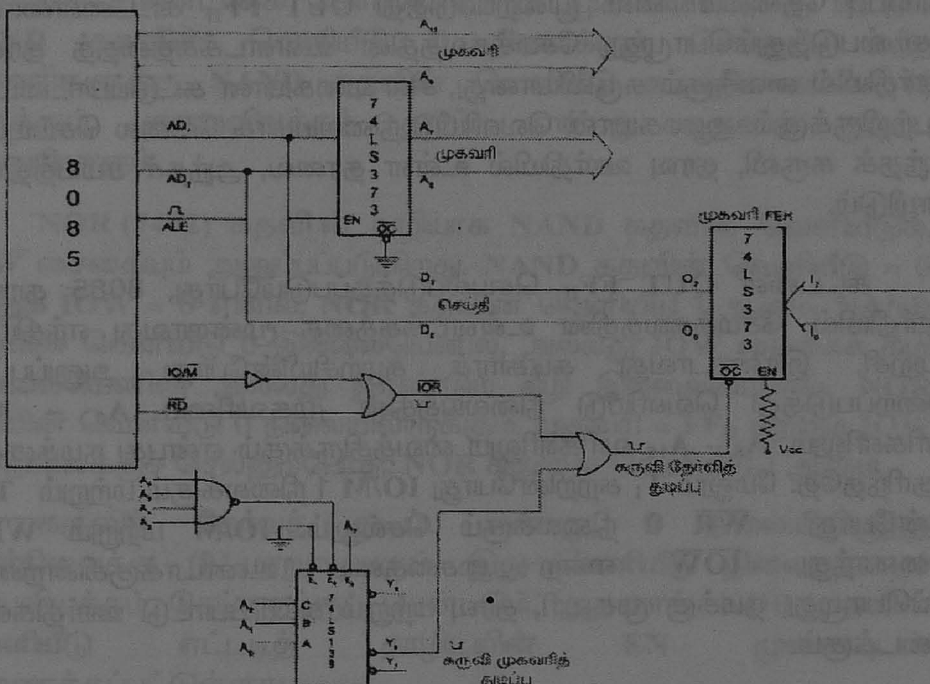
கருவியின் முகவரித் துடிப்பை 3–8 குறியீட்டுப் பெயர்ப்பியைக் கொண்டு, **FF$_H$** முகவரிக்காக உருவாக்கலாம். முகவரி **FE$_H$**க்கான ஈரடி சேர்ந்த அமைப்பு, கீழே காட்டப்பட்டுள்ளது.

A7	A6	A5	A4	A3	A2	A1	A0
1	1	1	1	1	1	1	0

இந்த முகவரி பிட்டுகளுடன், 3–8 குறியீட்டுப் பெயர்ப்பியை ஒரு வழியில் பயன்படுத்தும் முறை கீழே கொடுக்கப்பட்டுள்ளது.

E_3 உடன் A_7 இணைக்கவேண்டும். நாம் எடுத்துக்கொண்ட முகவரிக்கு 0 கொடுப்பதற்காக A_6 A_5 A_4 A_3 உள்ளீடு பிட்டுகள் ஒரு நான்கு உள்ளீடு **NAND** கதவுடன் சேர்க்கப்படும். **NAND** கதவின் வெளியீடு E_2 உடன் இணைக்கப்பட்டுள்ளது. மற்றொரு செயலியக்க முனை $\overline{E_1}$ ஐ 0ல் நிறுத்த, அந்தமுனையைத் தரையிட வேண்டும்.

A_2 A_1 மற்றும் A_0 பிட்டுகள், குறியீட்டுப் பெயர்ப்பியின் C B A உள்ளீடுகளில் இணைக்கப்படும். மூன்று செயலியக்க உள்ளீடுகளும், சரியான செயலியக்கம் கொண்டவை. உள்ளீடு = 110 எனில், வெளியீடு Y_6 மட்டும் 0 நிலைக்குச் செல்லும். இது கருவி முகவரித் துடிப்பாகப் பயன்படும். இந்தத் துடிப்புடன், $\overline{IOR}$ சைகையையும் ஒரு கதவில் கொடுக்கக் குறைவியக்கத் தேர்வித் துடிப்பொன்று உள்ளீடு நிலையத்தைச் செயலியக்கம் செய்வதற்காகத் தோன்றும். இந்த அமைப்பு, படம் 2.9இல் காட்டப்பட்டுள்ளது.



படம் 2.9 எட்டடி தாழிடல் மற்றும் குறியீட்டுப் பெயர்ப்பியைப் பயன்படுத்தி உள்ளீடு நிலைய இடைப்பரிமாறல்

எட்டுக் குறைவியக்க வெளியீடுகளைக் கொண்டு, எட்டு உள்ளீடு/வெளியீடு நிலையங்களைத் தேர்வு செய்வதற்காகக் குறியீட்டுப் பெயர்ப்பியைச் சக்தி வாய்ந்ததாகப் பயன்படுத்தலாம்.

2.4 வெளியீடு நிலையத் திட்டம் :

இந்தக் கருவியானது, எட்டு உள்ளீடுகளையும் எட்டு வெளியீடுகளையும் பெற்றிருக்கும். இந்தக் கருவியின் உள்ளீடு 8085இன் தரவு ஊர்தியுடன் இணைக்கப்படும். வெளியீடு நிலையமானது, தரவு ஊர்தியிலிருந்து தரவை மட்டும் சேகரிக்கும். தரவுத் தாழிடல் வாய்ப்பு மட்டும், அதற்குத் தேவை. வெளியீடு நிலையத்தின் வெளியீடு **LED** அஞ்சல், **A/D** மாற்றி போன்ற கருவிகளுடன் இணைக்கப்படும். அதனால் உயர் நடைதடை வாய்ப்பு தேவையில்லை. முறைப்படுத்தி **OUT** $\mathbf{FF_H}$ கட்டளையைச் செயல்படுத்தும்பொழுது, சேமக்கலத்தின் உள்ளடக்கத்தைத் தரவு ஊர்தியில் வைக்கும் கருவியானது, சில வகையான கட்டுப்பாட்டைப் பெற்றிருக்கும். ஆகையால், வெளியீடு நிலையமாக வேலை செய்யும். அந்தக் கருவி, தரவு ஊர்தியில் உள்ள தரவை, அந்தச் சமயத்தில் தாழிடும்.

கட்டளை **OUT** $\mathbf{FF_H}$ செயல்படுத்தப்படும்போது, 8085 தரவு ஊர்தியில், சேமக்கலத்தின் உள்ளடக்கத்தை மூன்றாவது எந்திரச் சுழற்சி, இரண்டாவது கடிகாரச் சுழற்சியின்போது அனுப்பும். முறைப்படுத்தி வெளியீடு நிலையத்தின் முகவரியை $A_0 - A_7$ வரிகளிலும், $A_8 - A_{15}$ வரிகளிலும் வைத்திருக்கும் என்பது நமக்குத் தெரிந்ததே. மேலும் T_1 சுற்றின்போது $IO/\overline{M}$ 1 நிலைக்கும் மற்றும் T_2 இன்போது $\overline{WR}$ 0 நிலைக்கும் செல்லும். $IO/\overline{M}$ மற்றும் $\overline{WR}$ இணைந்து, $\overline{IOW}$ என்ற சைகையை உண்டாக்குகின்றன. இப்பொழுது, நமக்கு முகவரி, தரவு மற்றும் கட்டுப்பாட்டு ஊர்திகள் கிடைக்கும்.

$\mathbf{FF_H}$ ன் 8 பிட் முகவரியையும் $\overline{IOW}$ சைகைகளையும் சேர்ந்த்தொரு உயரியக்கக் கருவி, தேர்வித் துடிப்பை உருவாக்கும். இந்தத் துடிப்பு எட்டடித் தாழிடலின் **EN** கட்டுப்பாட்டில் கொடுக்கப்படும்.

$\mathbf{FF_H}$ இன் முகவரியை ஈரடியாக மாற்றும்பொழுது எட்டு 1–கள் கிடைக்கும். முகவரியை எடுத்துச்செல்லும் முகவரி வரிகள், கீழே கொடுக்கப்பட்டுள்ளன.

A7	A6	A5	A4	A3	A2	A1	A0
1	1	1	1	1	1	1	1

அனைத்து எட்டு முகவரி பிட்டுகளும், 8 உள்ளீடு NAND கதவிற்கு உள்ளீடாகக் கொடுக்கப்படும். முகவரி FF_Hக்கு மட்டும் NAND கதவின் வெளியீடு 0 நிலைக்குச் செல்லும். மற்ற முகவரிகளுக்கு NAND கதவின் வெளியீடு உயர்நிலையிலே (1) இருக்கும். குறைவியக்கத் துடிப்பு என்பது கருவித் துடிப்பு முகவரியாகும்.

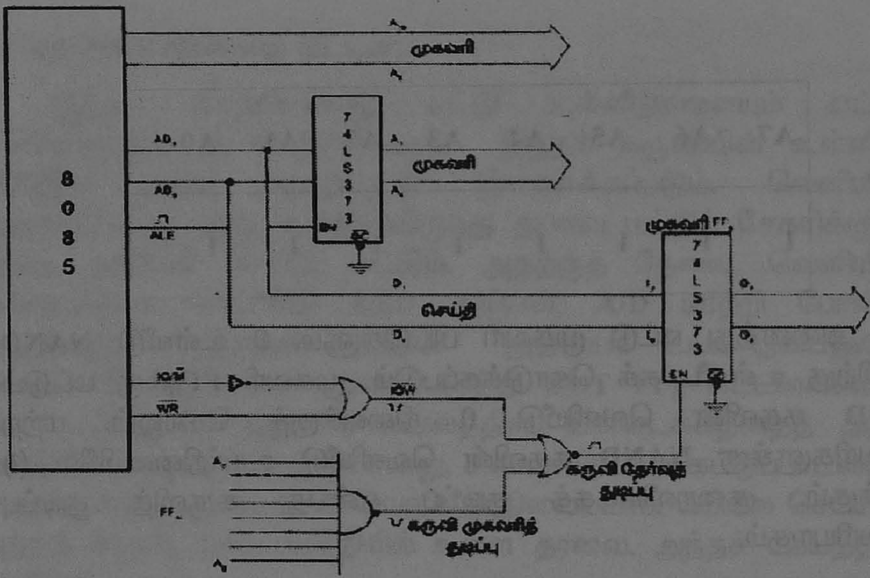
NOR (7402) கதவின் வழியாக NAND கதவின் வெளியீடும், $\overline{IOW}$ சைகையும் அனுப்பப்படுகிறது. NAND கதவின் வெளியீடு = 0 மற்றும் $\overline{IOW}$ = 0 எனில், NOR கதவின் வெளியீடு 1 ஆகும். NAND கதவின் வெளியீடு 0 இல்லையெனில், அல்லது $\overline{IOW}$ சைகைச் சுழி இல்லையெனில், அல்லது இரண்டும் சுழி இல்லையெனில், NOR கதவின் வெளியீடு 0 நிலையிலிருக்கும். முகவரி = FF_H மற்றும் $\overline{IOW}$ = 0 நிலைக்குச் செல்லும்போது, NOR கதவின் வெளியீடு 1 ஆகும்.

கருவித் தேர்வுத் துடிப்பு என்பது, 0 நிலையிலிருந்து உயர்நிலைக்கு (1) மாறுவதாகும். இது வெளியீடு நிலையத்தைச் செயலியக்கம் செய்வதற்குப் பயன்படும். அதனால், NOR கதவின் வெளியீடு எட்டடித் தாழிடலின் EN முனையுடன் இணைக்கப்பட்டுள்ளது.

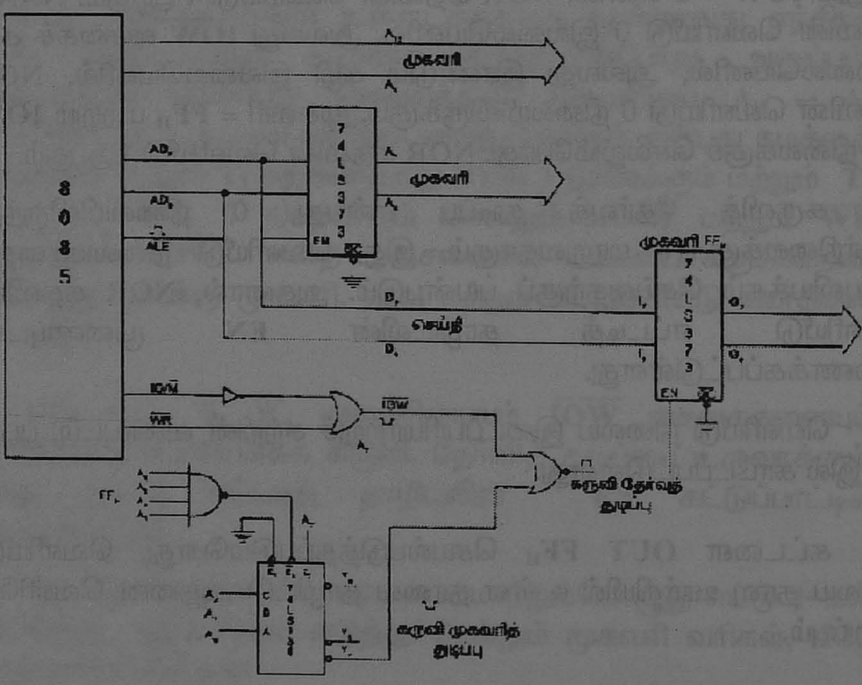
வெளியீடு நிலைய இடைப்பரிமாற்றுச் சுற்றின் வரைபடம், படம் 2.10இல் காட்டப்பட்டுள்ளது.

கட்டளை OUT FF_H செயல்படுத்தப்படும்போது, வெளியீடு நிலைய தரவு ஊர்தியில் உள்ள தரவை தாழிட்டு அதனை வெளியே அனுப்பும்.

உள்ளீடு நிலையத்தில் உள்ளதுபோல், 3–8 குறியீட்டுப் பெயர்ப்பியைப் பயன்படுத்தி, FF_H முகவரிக்காக முகவரித் தேர்வுத் துடிப்பை உண்டாக்கலாம்.



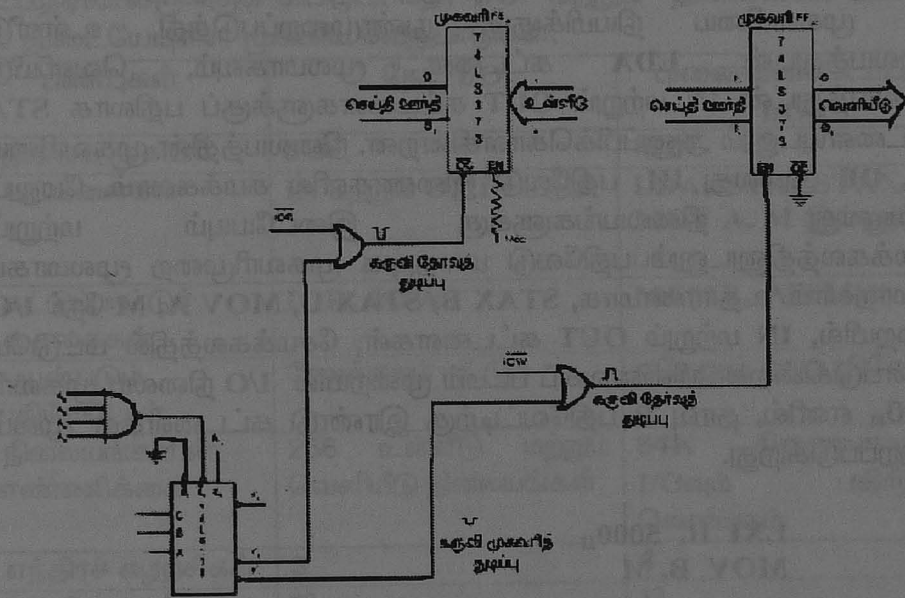
படம் 2.10 எட்டடித் தாழிடலைப் பயன்படுத்தி வெளியீடு துறைமுகம்



படம் 2.11 எட்டடித் தாழிடல் மற்றும் 3–8 குறியீட்டுப் பெயர்ப்பியைப் பயன்படுத்தி வெளியீடு நிலைய இடைப்பரிமாறல்

உள்ளீடு நிலையத்திற்குப் பயன்படுத்திய அதே மாதிரியான இணைப்புகளை, நாம் பயன்படுத்தலாம். உள்ளீடு 1 1 1 எனில், வெளியீடு Y_7 தாழ்நிலைக்குச் செல்லும். இது கருவி முகவரி துடிப்பாகப் பயன்படுகிறது. இது **NOR** கதவின் **IOW** உடன் இணைந்து, வெளியீடு செயலியக்கம் பெற, கருவித் தேர்வுச் சைகையை உண்டாக்குகிறது. இதன் அமைப்புப் படம் 2.11இல் காட்டப்பட்டுள்ளது.

உள்ளீடு மற்றும் வெளியீடு நிலையச் சுற்றுகள் இணைக்கப்பட்டு, ஒரு தனிப்பட்ட சுற்றாகப் படம் 2.12இல் காட்டப்பட்டுள்ளது.



படம் 2.12 உள்ளீடு மற்றும் வெளியீடு நிலையங்களுக்கான இணைந்த சுற்று

இந்த எளியச் சுற்றில், உள்ளீடு நிலையத்தின் முகவரி FE_H ம் வெளியீடு நிலையத்தின் முகவரி FF_H ம் உள்ளன. இந்தச் சுற்றை எட்டு உள்ளீடு நிலையங்கள் அல்லது எட்டு வெளியீடு நிலையங்கள் அல்லது சில உள்ளீடு நிலையங்கள் மற்றும் சில வெளியீடு நிலையங்கள் எனச் சுலபமாக மாற்றியமைக்கலாம். ஒவ்வொரு நிலையமும் வேறுபட்ட முகவரியைப் பெற்றிருக்கும். உதாரணமாக, $F8_H$, $F9_H$, FA_H, FB_H, FC_H, FD_H, FE_H மற்றும் FF_H.

நினைவகம் – படமம் I/O (Memory – Mapped I/O) :

இதுவரை விவாதிக்கப்பட்ட உள்ளீடு மற்றும் வெளியீடு இடைப்பரிமாற்றுச் சுற்றுகளுக்கு நேர் **I/O**, எளிய **I/O**, புறவியலான **I/O**அல்லது **I/O** படமம் **I/O** என்று பெயர். மற்றொரு முறையிலும், உள்ளீடு நிலையங்கள் மற்றும் வெளியீடு நிலையங்கள் இடைப்பரிமாறல் செய்யலாம். இதற்கு நினைவகப் படமம் **I/O** என்று பெயர். நினைவகப் படமம் **I/O** முறையில், **8085** உள்ளீடு நிலையத்தை, நினைவக இருப்பிடம் போன்று நோக்குகிறது. நிலையங்களின் முகவரிகள் நியமிக்கப்பட்டவை என்பதால், **ROM** அல்லது **RAM** ஆல் பயன்படுத்த முடியாது. உள்ளீடு நிலையம், **16** பிட் முகவரியை நியமிக்கும். நுண்முறைப்படுத்தி, உள்ளீடு நிலையத்துடன் **LDA** கட்டளை மூலமாகவும், வெளியீடு நிலையத்துடன் **IN** மற்றும் **OUT** கட்டளைகளுக்குப் பதிலாக **STA** கட்டளையுடனும் அனுப்பிக்கொள்கின்றன. நிலையத்தின் முகவரியை **BC, DE** அல்லது **HL** பதிவேடு இணைகளில் சுமக்கலாம். மேலும் தரவானது நிலையங்களுக்கு இடையேயும் மற்றும் சேமக்கலத்தினுடனும் பதிவேடு மறைமுக முகவரிமுறை மூலமாகப் பரிமாறலாம். உதாரணமாக, **STAX B, STAX D, MOV A, M** நேர் **I/O** முறையில், **IN** மற்றும் **OUT** கட்டளைகள், சேமக்கலத்தில் மட்டுமே பயன்படுகின்றன. பினைவகப் படமம் முறையில் **I/O** நிலைய முகவரி **5000_H** எனில், தரவு **B** பதிவேட்டிற்கு இரண்டு கட்டளைகள் மூலம் மாற்றப்படுகிறது.

LXI H, 5000_H
MOV B, M

இதேபோல், வேறு பல பதிவேடுகளையும் பயன்படுத்தலாம். நிலைய முகவரியின் தொடக்கத்தை **LXI H, 5000_H** எனக்கொண்டால், **ADD M** என்பது சேமக்கலத்திலுள்ளவற்றுடன் உள்ளீடு நிலையத்தின் தரவையும் கூட்டும். **ORA M** கட்டளை, வாதியல் **OR** செயற்பாட்டைச் சேமக்கலம் மற்றும் உள்ளீடு நிலையத்திலுள்ள தரவு, இரண்டுக்குமிடையே நிறைவேற்றும்.

நினைவகச் சம்மந்தப்பட்ட கட்டளைகள் பயன்படுத்தப்படுவதால், கட்டுப்பாட்டுச் சைகைகளும், MEMR மற்றும் MEMW ஆகப் பயன்படுத்தப்படுகின்றன. உள்ளீடு அல்லது வெளியீடு இடைப்பரிமாற்று திட்டம் வரையும்போது, அனைத்து 16 முகவரி வரிகளும் குறியீடு பெயர்க்கப்பட்டு, கருவி முகவரித் துடிப்பாகக் கிடைக்கும்.

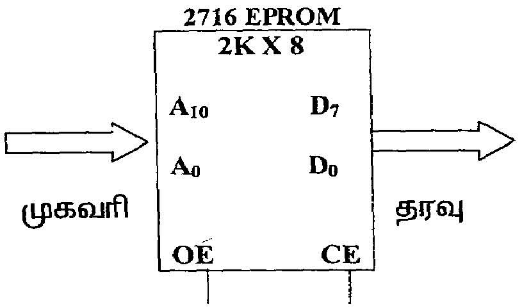
ஆதலால், நேர் I/O முறை எளியது. ஆனால், IN மற்றும் OUT கட்டளைகளில் மட்டுமே பயன்படுகின்றன. பினைவகப் படமம் I/O முறையில், முகவரிக் குறியீட்டுப் பெயர்ப்புச் சுற்றுக்காக, நிறைய வன்மியங்கள் தேவைப்படுகின்றன. ஆனாலும், நிறைய கட்டளைகள் திட்டநிரலாளருக்குக் கிட்டும். நேர் I/O மற்றும் நினைவகப் படமம் I/O இடையேயுள்ள முக்கிய வேறுபாடுகள்.

பண்புகள்	நேர் I/O	பினைவகப் படமம் I/O
1. நிலைய முகவரி	8 பிட்	16 பிட்
2. கட்டளைகள்	IN மற்றும் OUT மட்டும்	LDA, STA, STAX B MOV A, M இன்னும் பல
3. கட்டுப்பாட்டுச் சைகைகள்	IOR, IOW	MEMR, MEMW
4. பயன்படும் பதிவேடுகள்	சேமக்கலம் மட்டும்	ஏதேனும் ஒரு பதிவேடு
5. நிலையங்களின் எண்ணிக்கை	256 உள்ளீடு மற்றும் வெளியீடு நிலையங்கள்	64K நினைவகமும், I/Oவும் பகிர்ந்து கொள்ளும்.
6. எந்திரச் சுழற்சிகள்	3	4
7. T நிலைகள்	10	13
8. வன்மியம்	8 பிட் முகவரியைக் குறியீட்டுப் பெயர்த்தல். எளிதாக நிறைவேற்ற முடியும்.	16 பிட் முகவரியைக் குறியீட்டுப் பெயர்த்தல். கூடுதல் சுற்றுகள் தேவைப்படும்.
9. மென்மியம்	IN மற்றும் OUT கட்டளைகளில் மட்டும் பயன்படும்.	கணிதவியல் மற்றும் வாதியல் செயற்பாடுகளை நிறைவேற்ற முடியும்.

2.6 ROM / EPROM இடைப்பரிமாறல் :

a) 2K X 8 EPROM இடைப்பரிமாறல் :

2K X 8 ROM இன் அடையாள வரைபடம், 2.13இல் காட்டப்பட்டுள்ளது.



படம் 2.13 2 K X 8 EPROM

(2K X 8) ROMஇல் 2048 நினைவக இருப்பிடங்களுள்ளன. ஒவ்வொரு இருப்பிடமும் 8 பிட் தரவைச் சேமிக்கும். 2048 இருப்பிடங்களை முகவரியிட, 11 முகவரி வரிகள் தேவைப்படுகின்றன. குறைவியக்கச் சில் தேர்வி ($\overline{CS}$) அல்லது சில் செயலியக்கச் ($\overline{CE}$) சைகையுடன் சில்லைத் தேர்வு செய்த பிறகு, $\overline{RD}$ தாழ் நிலையிலாக்கி நினைவக இருப்பிடத்திலுள்ள தரவைப் படிக்க முடியும். நிறைய எண்ணிக்கையிலுள்ள சில்களிலிருந்து ஒரு குறிப்பிட்ட சில்லை மட்டும் தேர்வு செய்வதற்கு சில் சைகைத் தேர்வி தேவைப்படுகிறது. உயர் வரிசை முகவரி வரிகள், $\overline{CE}$ சைகையை உண்டாக்குவதற்குப் பொதுவாகப் பயன்படும். ஒரு கொடுக்கப்பட்ட முகவரிக்கு, $\overline{CE} = \overline{OE} = 0$ எனில், அந்த நினைவக முகவரி இருப்பிடத்திலுள்ள தரவானது தரவு வரிகளில் (D_O – D_7) விழும். ஏதாவது ஒன்று உயர்நிலையிலிருந்தால், தரவு வரிகள் உயர் நடைதடை நிலைக்குச் செல்லும். ஒருமுறை வெளியீடு தரவு வரிகள் உயர் நடைதடை நிலை அல்லது மும்மை நிலையிலிருந்தால், அமைப்பிலிருந்து சில் தனிமையாக்கப்படும் அல்லது துண்டிக்கப்படும். இந்த அமைப்பு முறை, மிக முக்கியமானது. ஏனெனில், அனைத்து நினைவகச் சில்களிலுள்ள தரவு வரிகள் மற்றும் I/O கருவிகள், ஒரு பொதுவான தரவு ஊர்தியுடன் இணைக்கப்பட்டுள்ளன.

அடுத்தபடியாக, நாம் இடைப்பரிமாறல் தேவைக்கு ஏற்றாற்போல், **ROM** சில்லிற்கான சரியான முகவரியைத் தீர்மானிக்க வேண்டும். ஒரு நுண்கணிப்பொறி அமைப்பில், எண்ணற்ற **ROM** மற்றும் **RAM** சில்களைப் பயன்படுத்தலாம். அவை ஒவ்வொன்றும், ஒரு குறிப்பிட்ட முகவரி நெடுக்கத்தைக் கொண்டிருக்கும். கொடுக்கப்பட்ட முகவரி ஒன்றிற்கு, ஒரே ஒரு சில்லை மட்டும் தேர்வு செய்ய வேண்டும். சரியான முகவரி நெடுக்கத்தை **RAM** மற்றும் **ROM**ற்குப் பங்கிடுவதற்கு நினைவகப் படமம் என்று பெயர்.

நுண்கணிப்பொறியை முடுக்கிவிடும்போது, 8085 முறைப்படுத்தியிலுள்ள திட்டநிரல் எண்ணி, 0000_H ஐப் படிக்கும். இது எதனாலென்றால், மின்திறன் வழங்கலை முடுக்கிவிட்ட பிறகும், 8085இன் $\overline{\text{RESET+IN}}$ முனை 0 நிலையில் சில மில்லி வினாடிகளுக்கு வைக்கப்படும். இதற்காகத் திறன்மேல் மறுஅமைவுச்சுற்று பயன்படுகிறது. அந்த முகவரியிலிருந்து கட்டளைகள் எடுக்கப்பட்டு, ஒவ்வொன்றாகச் செயல்படுத்தப்படுகின்றன. ஆகையால், **(2K X 8)** கொள்ளளவுடைய **ROM**இன் தொடக்க முகவரி 0000_H. இது 2048 இருப்பிடங்களைப் பெற்றிருக்கும். இதன் கடைசி முகவரி **($07FF_H$)** .

2K X 8 ROM முகவரிக்கு, 11 முகவரி வரிகள் மட்டும் தேவைப்படுகின்றன. ஆனால் 8085 எப்பொழுதும் 16 பிட் முகவரியை அனுப்புகிறது. ஆகவே, 8085இன் முகவரி வரிகள், $A_0 - A_{10}$ நினைவகச் சில்லிலுள்ள $A_0 - A_{10}$ முகவரி வரிகளுடன் இணைக்கப்பட வேண்டும். 8085இன் பெரும முக்கியத்துவ **(MSB)** ஐந்து முகவரி வரிகள், $A_{11} - A_{15}$ சில் தேர்விச் சைகையை உண்டாக்குவதற்குப் பயன்படுகின்றன. 8085இன் தரவு வரிகள் $D_0 - D_7$ **ROM**இன் தரவு வரிகளுடன் இணைக்கப்படும். நாம் உண்டாக்கிய $\overline{\text{MEMR}}$ நினைவகப் படி சைகை **ROM**இன் $\overline{\text{OE}}$ உடன் இணைக்கப்படும். இப்பொழுது, 8085இன் முகவரி ஊர்தி, தரவு ஊர்தி மற்றும் கட்டுப்பாட்டு ஊர்தி **ROM**இன் முகவரி, தரவு மற்றும் கட்டுப்பாட்டு வரிகளுடன் இணைக்கப்படுகின்றன. நாம் சில் தேர்விச் சைகையை உண்டாக்கி, அதனை **ROM**இன் $\overline{\text{CE}}$ உள்ளீட்டுடன் இணைத்து, இடைப்பரிமாறல் சுற்றின் இணைப்பை நிறைவு செய்ய வேண்டும்.

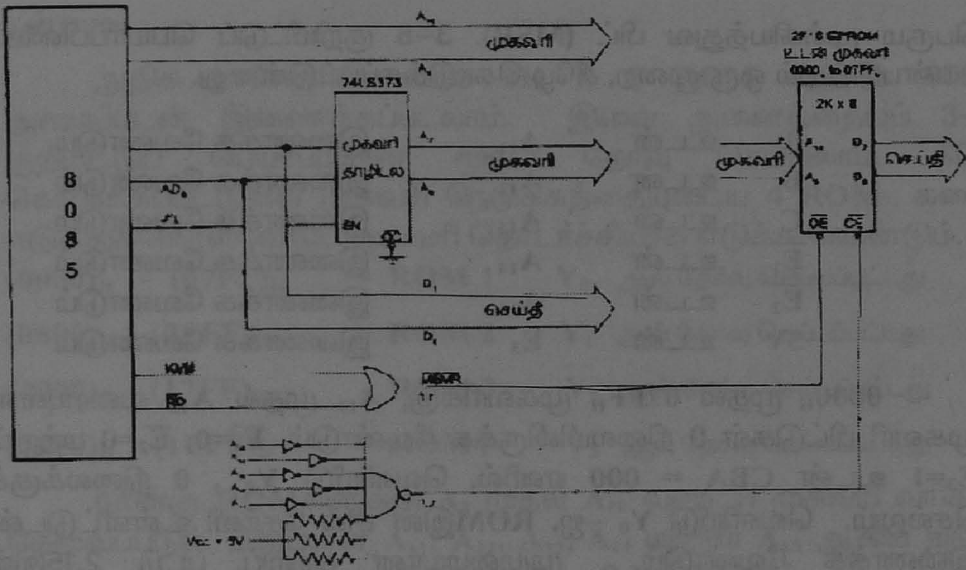
ROMஇன் முகவரி நெடுக்கத்தை **(0000_H – $07FF_H$)**, ஈரடி முறையில் எழுதலாம்.

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	பதினாறடி
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000_H
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	$07FF_H$

முதல் இருப்பிடத்தின் முகவரி 0000_H
இறுதி இருப்பிடத்தின் முகவரி $07FF_H$

முகவரி நெடுக்கம் முழுவதும், முகவரி பிட்டுகள் A_{11} முதல் A_{15} வரை, 0 நிலையிலுள்ளன. இந்த ஐந்து சுழிகள், சில் தேர்விச் சைகை உண்டாக்குவதற்குப் பயன்படுகின்றன. 8 உள்ளீடு NAND கதவை, முதலில் நாம் பயன்படுத்தலாம். முகவரி வரிகள் (A_{11} – A_{15}) ஐந்தும் நேர்மாறாக்கப்பட்டு, 8 உள்ளீடு NAND கதவின் ஏதாகிலும் ஐந்து உள்ளீடுகளுடன் இணைக்கப்படும். NAND கதவின் மீதமுள்ள மூன்று உள்ளீடுகளும் உயர்நிலையில் வைப்பதற்கு, அம்முனைகள் பாதுகாப்பான மின் தடையினால் 5V மின்திறன் மின்கலத்துடன் இணைக்கப்படுகின்றன. A_{11} முதல் A_{15} வரையுள்ள ஐந்து பிட்டுகள் 0 நிலையிலிருக்கும்போது, NAND கதவின் எல்லா உள்ளீடுகளும் 1 நிலையை அடையும். ROM முகவரி 0000_H - $07FF_H$ ஆக இருக்கும்போது, NAND கதவின் வெளியீடு 0 நிலைக்குச் செல்லும். ஆகையால், NAND கதவின் வெளியீட்டை ROM சில்லைத் தேர்வு செய்வதற்குப் பயன்படுத்தலாம். நெடுக்கத்திற்கு அப்பால் இது இருந்தால், NAND கதவின் வெளியீடு 1 ஆகும். அதனால் ROM தேர்வு செய்யப்படாது.

உதாரணமாக, கொடுக்கப்பட்ட முகவரி 0800_H எனில், A_{15}, A_{14}, A_{13}, A_{12}, A_{11} பிட்டுகளின் சேர்க்கை 0001 ஆகும். A_{11} பிட் 1 எனில், அதனைத் தலைகீழாக்கும்போது 0வாகும். NAND கதவில் ஏதாவதொரு உள்ளீடு 0 நிலையிலிருந்தால், அதன் வெளியீடு 1 நிலைக்குச் செல்லும். மேலும் கொடுக்கப்பட்ட முகவரி 0800_H ஆக இருந்தால், 0000_H – $07FF_H$ முகவரி நெடுக்கம் உடைய ROM சில் தேர்வு செய்யப்பட மாட்டாது.



படம் 2.14 2KX8 EROM இடைப்பரிமாறல்

2.14 2K X 8 EROM இடைப்பரிமாறலில் காட்டப்பட்டுள்ளது. இது **2K X 8 EPROM**-ற்கான சுற்று. சில் தேர்விச் சைகையை 8 உள்ளீடு **NAND** கதவைப் பயன்படுத்திப் பெறுவதற்குப் பதிலாக, **3–8** குறியீட்டுப் பெயர்ப்பியைக்கூடப் **(74LS138)** பயன்படுத்தலாம். இன்டெல் ஸ்தாபனத்தால் கண்டு பிடிக்கப்பட்ட மற்றொரு சில் **8205**ஐகூடப் பயன்படுத்தலாம். இரண்டு சில்கள் ஒரே மாதிரியான வேலையைச் செய்யும்.

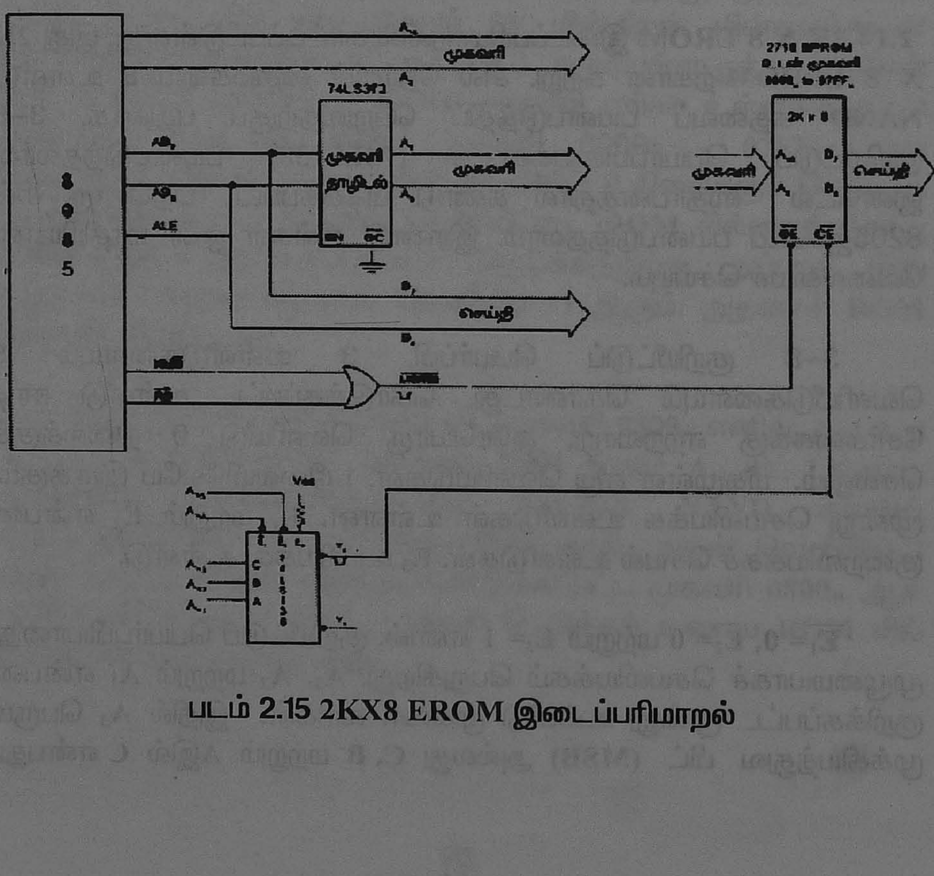
3–8 குறியீட்டுப் பெயர்ப்பி, 3 உள்ளீடுகளையும் 8 வெளியீடுகளையும் கொண்டது. கொடுக்கப்பட்ட உள்ளீடு ஈரடி சேர்க்கைக்கு ஏற்றவாறு, ஒரேயொரு வெளியீடு 0 நிலைக்குச் செல்லும். மீதமுள்ள ஏழு வெளியீடுகள், 1 நிலையிலேயே இருக்கும். மூன்று செயலியக்க உள்ளீடுகள் உள்ளன. $\overline{E_1}$ மற்றும் $\overline{E_2}$ என்பன குறைவியக்கச் செயல் உள்ளீடுகள். E_3 உயரியக்க உள்ளீடு.

$\overline{E_1} = 0$, $\overline{E_2} = 0$ மற்றும் $E_3 = 1$ எனில், குறியீட்டுப் பெயர்ப்பியானது முழுமையாகச் செயலியக்கம் பெறுகிறது. A_3, A_2 மற்றும் A_1 என்பன குறிக்கப்பட்ட மூன்று உள்ளீடு முகவரி வரிகள். இதில் A_3 பெரும முக்கியத்துவ பிட் **(MSB)** அல்லது **C, B** மற்றும் **A**இல் **C** என்பது

பெரும முக்கியத்துவ பிட் (MSB). 3–8 குறியீட்டுப் பெயர்ப்பியைப் பயன்படுத்தும் ஒரு முறை, கீழே கொடுக்கப்பட்டுள்ளது.

A	உடன்	A_{11}	இணைக்க வேண்டும்
B	உடன்	A_{12}	இணைக்க வேண்டும்
C	உடன்	A_{13}	இணைக்க வேண்டும்
$\overline{E_1}$	உடன்	A_{14}	இணைக்க வேண்டும்
$\overline{E_2}$	உடன்	A_{15}	இணைக்க வேண்டும்
5V	உடன்	E_3	இணைக்க வேண்டும்

0000_H முதல் $07FF_H$ முகவரிக்கு, A_{11} முதல் A_{15} வரையுள்ள முகவரி பிட்டுகள் 0 நிலையிலிருக்க வேண்டும். $\overline{E_1}=0$, $\overline{E_2}=0$ மற்றும் $E_3=1$ உடன் CBA = 000 எனில், வெளியீடு Y_0 , 0 நிலைக்குச் செல்லும். வெளியீடு Y_0 –ஐ, ROMஇன் சில் தேர்வி உள்ளீட்டுடன் இணைக்க வேண்டும். முழுமையான சுற்றுப் படம் 2.15இல் காட்டப்பட்டுள்ளது.



படம் 2.15 2KX8 EROM இடைப்பரிமாறல்

நன்மை :

அதிக எண்ணிக்கையில் **2K X 8 ROM** சில்கள் 8085 அமைப்புடன் இணைக்கப்படலாம். இவை அனைத்தையும் 3–8 குறியீட்டுப் பெயர்ப்பியின் மூலம் தேர்வு செய்யலாம். கீழே கொடுக்கப்பட்டுள்ள முகவரி நெடுக்கத்தையுடைய 4 **ROM** களை எடுத்துக்கொள்வோம். முகவரி தொடர்ச்சியாக எடுக்க வேண்டும்.

$(0000)_H$ - $(07FF)_H$ - ROM 1 **Y_0** ஆல் தேர்வு செய்யப்பட்டது

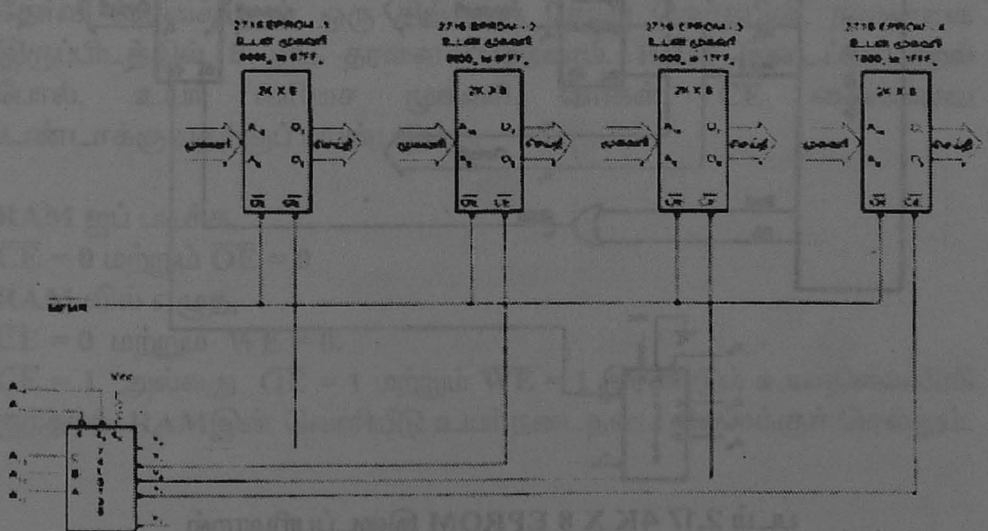
$(0800)_H$ – $(0FFF)_H$ - ROM 2 **Y_1** ஆல் தேர்வு செய்யப்பட்டது

$(1000)_H$ – $(17FF)_H$ - ROM 3 **Y_2** ஆல் தேர்வு செய்யப்பட்டது

$(1800)_H$ - $(1FFF)_H$ - ROM4 **Y_3** ஆல் தேர்வு செய்யப்பட்டது

நான்கு **ROM**களுக்கு A_0 முதல் A_{10} வரை 11 முகவரி வரிகள் தேவைப்படும். மீதமுள்ள A_{11}, A_{12}, A_{13}, A_{14} மற்றும் A_{15} ஆகிய ஐந்து வரிகளும், ஒவ்வொரு சில்லிற்கும் சில் தேர்விச் சைகையை உண்டாக்குவதற்கு பயன்படும்.

முகவரி வரிகள் A_0 முதல் A_{10} வரை, எல்லா சில்களுக்கும் பொதுவானது. மேலும், அவை இணையாக இணைக்கப்பட்டுள்ளன. அதேபோல் D_0 முதல் D_7 வரை உள்ள தரவு வரிகளும், இணையாக இணைக்கப்படலாம். இந்த அமைப்புப் படம் 2.16இல் காட்டப்பட்டுள்ளது.



படம் 2.16 நான்கு **(2K X 8) EPROM** களை இடைப்பரிமாறல் செய்தல்.

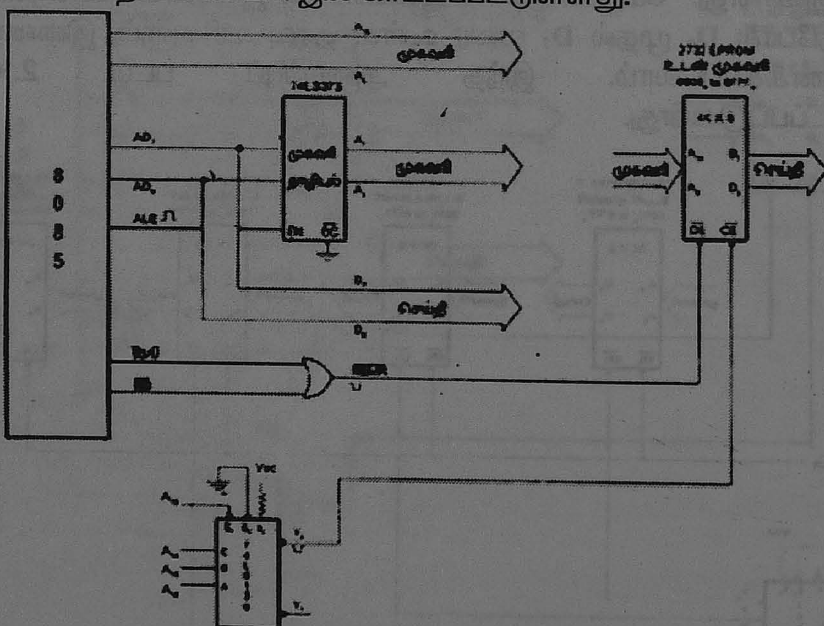
b) 4K X 8 ROM இடைப்பரிமாறல் :

ஒரு **4K X 8 ROM**, A_0 முதல் A_{11} வரை, 12 முகவரி வரிகளைப் பெற்றிருக்கும். ஆகையால், A_{11} முகவரி **ROM** உடன் இணைக்கப்படும். **4K X 8 ROM**இன் முகவரி நெடுக்கம், 0000_H முதல் $0FFF_H$ ஆகும். A_{14}, A_{13} மற்றும் A_{12} முறையே **C, B** மற்றும் **A** இன் உள்ளீடுகளுடன் இணைக்க வேண்டும். கட்டுப்பாட்டு உள்ளீடு E_3ஐ **5V** மின்கலத்துடன் இணைக்கவேண்டும். 0 நிலையில் வைப்பதற்காக $\bar{E}_2$ தரையிடப்பட வேண்டும். முகவரி வரி A_{15} ஐ $\overline{E_1}$ உடன் இணைக்க வேண்டும்.

முழுமையான முகவரி நெடுக்கத்திற்கு,

$$A_{15} \quad A_{14} \quad A_{13} \quad A_{12} \quad = \quad 0 \quad 0 \quad 0 \quad 0$$

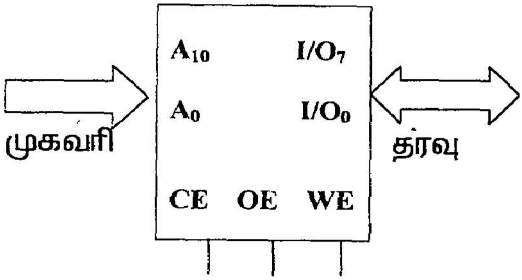
என வைக்க வேண்டும். குறியீட்டுப் பெயர்ப்பி வெளியீடு **C B A** முறையே 000 என வைக்க வேண்டும். ஆகையால், வெளியீடு Y_0 இப்பொழுது 0 வாகும். இது Y_0 சில்லைத் தேர்வு செய்வதற்குப் பயன்படும். தரவு ஊர்தி மற்றும் நினைவகப் படி சைகைகளின் இணைப்பில், எந்தவிதமான மாற்றமும் இல்லை. **4K X 8 ROM** இடைப்பரிமாறல் படம் 2.17இல் காட்டப்பட்டுள்ளது.



படம் 2.17 **4K X 8 EPROM** இடைப்பரிமாறல்

2.7 RAM இடைப்பரிமாறல் :

நுண்முறைப்படுத்தி 8085யுடன் **2K X 8 RAM (6116)**ஐ, இடைப்பரிமாறல் செய்யலாம். **2K X 8 RAM**இன் முனை வரைபடமானது, படம் 2.18இல் காட்டப்பட்டுள்ளது.

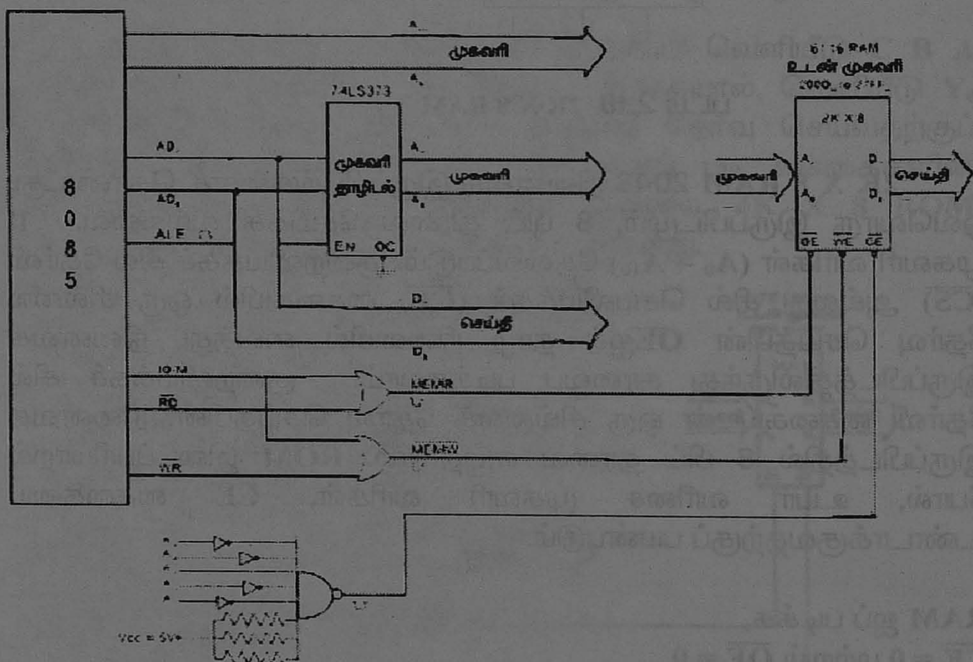


படம் 2.18 **2K X 8 RAM**

2K X 8 RAM 2048 நினைவக இருப்பிடங்களைக் கொண்டது. ஒவ்வொரு இருப்பிடமும், 8 பிட் தரவை சேமிக்க வல்லவை. 11 முகவரி வரிகள் (A_0 – A_{10}) தேவைப்படும். குறைவியக்கச் சில் தேர்வி ($\overline{CS}$) அல்லது சில் செயலியக்கச் ($\overline{CE}$) சைகையில் ஒரு சில்லை தேர்வு செய்தபின் $\overline{OE}$ஐத் தாழ் நிலையில் வைத்து, நினைவக இருப்பிடத்திலிருந்து தரவைப் படிக்கலாம். குறைவியக்கச் சில் தேர்வி சைகையுடன் ஒரு சில்லைத் தேர்வு செய்தபின், நினைவக இருப்பிடத்தில் 8 பிட் தரவை எழுதலாம். **ROM** இடைப்பரிமாறல் போல், உயர் வரிசை முகவரி வரிகள், $\overline{CE}$ சைகையை உண்டாக்குவதற்குப் பயன்படும்.

RAM ஐப் படிக்க,
$\overline{CE} = 0$ மற்றும் $\overline{OE} = 0$
RAM லில் எழுத,
$\overline{CE} = 0$ மற்றும் $\overline{WE} = 0$.
$\overline{CE} = 1$ அல்லது $\overline{OE} = 1$ மற்றும் $\overline{WE} = 1$ இரண்டும் உயர்நிலையிலிருந்தால், **RAM**இன் வெளியீடு உயர் நடைதடை நிலைக்குச் செல்லும்.

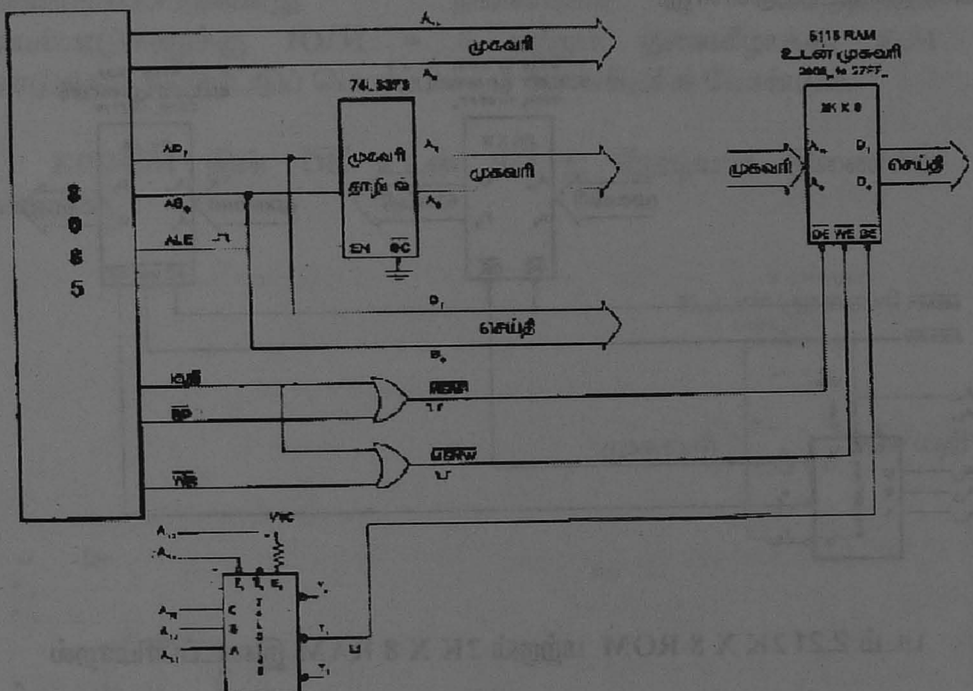
2K X 8 RAM-க்கான முகவரி நெடுக்கம் **2000_H** முதல் **$27FF_H$** வரை எனத் தேர்வு செய்யலாம். புதிய முகவரி நெடுக்கத்திற்குச் சில்லைக் கட்டாயமாகத் தேர்வு செய்ய வேண்டும். மேலும், நினைவக எழுது சைகை **$\overline{MEMW}$** ஐப் பயன்படுத்தி, **RAM**இன் **$\overline{WR}$** எழுதுக் கட்டுப்பாட்டைச் செயலியக்க வேண்டும். **2000_H** முதல் **$27FF_H$** வரையிலான முகவரி நெடுக்கத்திற்கான ஐந்து பெரும முக்கியத்துவப் பிட்டுகள் **A_{15}, A_{14}, A_{13}, A_{12}, A_{11} = 00100.** **A_{13}** முகவரி வரி 1 நிலையில் முன்பே இருப்பதால், **A_{13}**ஐ **NOT** கதவு இல்லாமல் **NAND** கதவுடன் நேரிடையாக இணைக்கலாம். **2000_H** முதல் **$27FF_H$** வரையுள்ள முகவரிக்கு **NAND** கதவின் வெளியீடு 0 ஆகும். **2KX8 RAM** இடைப்பரிமாறல் வரைபடம், படம் 2.19இல் காட்டப்பட்டுள்ளது.



படம் 2.19 **4K X 8 RAM** இடைப்பரிமாறல்

சில் தேர்விச் சைகையைப் பெறுவதற்கு, 8 உள்ளீடு **NAND** கதவைப் பயன்படுத்துவதற்குப் பதிலாக, நாம் 3–8 குறியீட்டுப் பெயர்ப்பி (**74LS138**)யைப் பயன்படுத்தலாம். ஆனால், இங்கு **A_{13}=1**

ஆதலால், **C B A = 1 0 0** மற்றும் **Y4** இன் வெளியீடு 0 நிலைக்குச் செல்லும். **Y4** வெளியீடை **RAM** சில்லின் தேர்வு உள்ளீட்டுடன் இணைக்க வேண்டும். மேலும், சைகை $\overline{\textbf{MEMW}}$ஐ **RAM**இன் $\overline{\textbf{WR}}$ உடன் இணைக்க வேண்டும். முழுமையான சுற்றுப் படம் 2.20இல் கொடுக்கப்பட்டுள்ளது.

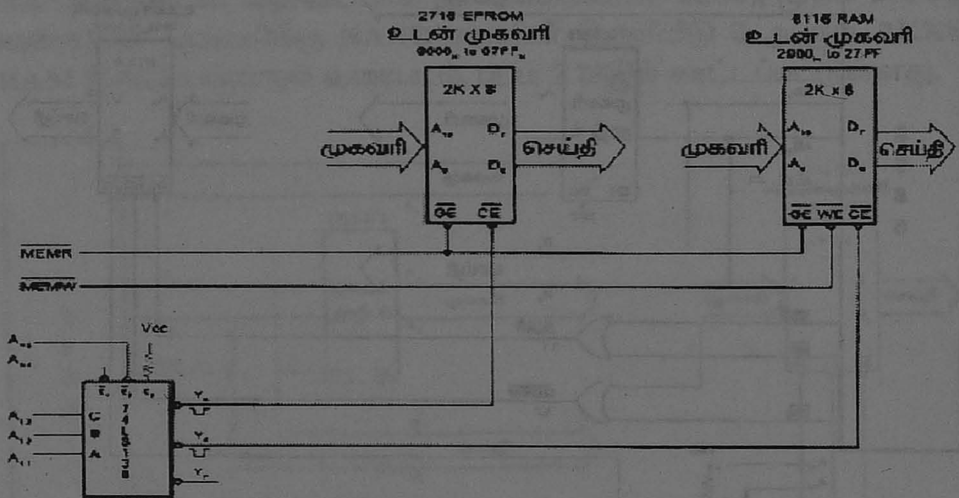


படம் 2.20 குறியீட்டுப் பெயர்ப்பியைப் பயன்படுத்திய 2K X 8 RAM இடைப்பரிமாறல்

மற்றொரு **2KX8 RAM** ஐ, தொடர்ச்சியான முகவரி **2800_H** **$2FFF_H$** உடன் நாம் சேர்த்துக் கொள்ளலாம்.

இதற்கான முகவரி **A_{15}, A_{14}, A_{13}, A_{12}, A_{11} = 00101** மற்றும் **C B A = 1 0 1.** ஆகையால், **Y_5** குறியீட்டுப் பெருக்கியின் வெளியீடு 0 நிலைக்குச் செல்லும். குறியீட்டுப் பெருக்கியின் வெளியீடு **Y_5**, இரண்டாவது **RAM** இன் சில் தேர்வி உள்ளீடுடன் இணைக்கப்படும்.

படம் 2.15இல் 2K X 8 ROM இடைப்பரிமாறல் கொடுக்கப்பட்டுள்ளது. மற்றும் படம் 2.20இல் 2K X 8 RAM இடைப்பரிமாறல் கொடுக்கப்பட்டுள்ளது. 3–8 குறியீட்டுப் பெயர்ப்பியைப் பயன்படுத்தி, இந்த இரண்டையும் இணைத்து ஒரு தனிச்சுற்றாக்கலாம். முகவரி ஊர்தி, தரவு ஊர்தி மற்றும் கட்டுப்பாட்டுச் சைகைகள், ROM மற்றும் RAMக்கு இணையாக இணைக்கப்பட வேண்டும். இந்த அமைப்பு, படம் 2.21இல் கொடுக்கப்பட்டுள்ளது.



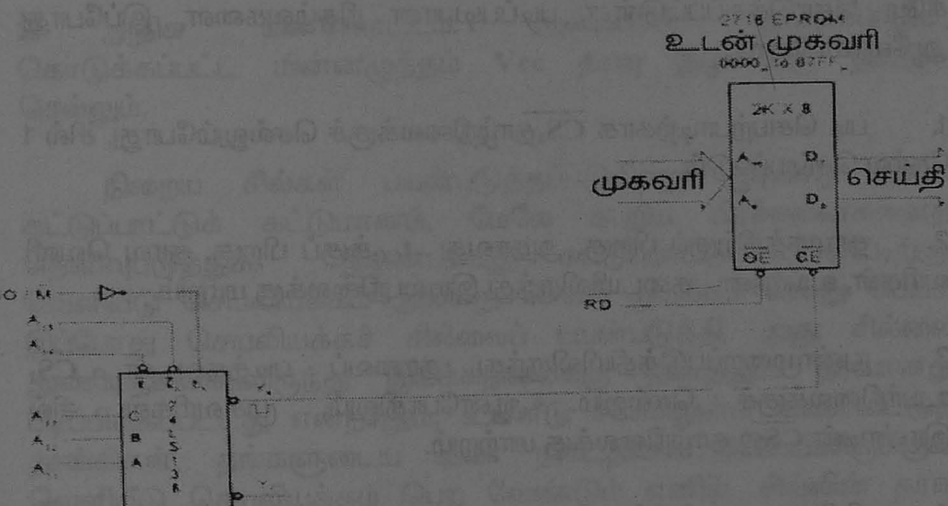
படம் 2.21 2K X 8 ROM மற்றும் 2K X 8 RAM இடைப்பரிமாறல்

இதுவரை நாம் பார்த்த அனைத்துச் சுற்றுகளிலும் நாம் **IO/$\overline{\text{M}}$**, **$\overline{\text{RD}}$** மற்றும் **$\overline{\text{WR}}$** ஆகியவற்றைச் சேர்த்து **$\overline{\text{MEMR}}$**, **$\overline{\text{MEMW}}$**, **$\overline{\text{IOR}}$** மற்றும் **$\overline{\text{IOW}}$** சைகைகளை உண்டாக்கினோம். கருவி முகவரி துடிப்பைப் பெறுவதற்கு, நாம் முகவரி வரிகளைப் பயன்படுத்தினோம். அவையாவும் நினைவகச் சில்லுடன் இணைக்கப்படவில்லை. நாம் **IO/$\overline{\text{M}}$** ஐ, **3-8** குறியீட்டுப் பெயர்ப்பியைப் பயன்படுத்தி, ஒரு கருவி முகவரித் துடிப்பைப் பெறலாம். நினைவகம் சம்மந்தமான செயல்களைச் செய்வதற்கு, **IO/$\overline{\text{M}}$** சைகை முன்பே பயன்படுத்தப்பட்டுள்ளது. **$\overline{\text{MEMR}}$**க்குப் பதிலாக **$\overline{\text{RD}}$**ஐ நேரடியாக பயன்படுத்தலாம். மேலும், **$\overline{\text{MEMW}}$**க்குப் பதிலாக **$\overline{\text{WR}}$**ஐ நேரடியாகப்

பயன்படுத்தலாம். இந்த விவாதத்தை அடிப்படையாகக் கொண்டு, **2K X 8 ROM** இடைப்பரிமாறலுக்கு ஒரு வேறுபட்ட சுற்றுப் படம் 2.22இல் கொடுக்கப்பட்டுள்ளது. முன்னதாகக் கொடுக்கப்பட்ட சுற்றுப் படத்துடன் **(2.14)** ஒப்பிட்டுப் பார்க்கும்போது, இரண்டு மாறுதல்கள் உள்ளன.

$IO/\overline{M}$ தலைகீழாக்கப்பட்டுக் குறியீட்டுப் பெயர்ப்பியுடன் இணைக்கப்பட்டுள்ளது. நினைவகச் சம்மந்தமான செயல்பாடுகளுக்கு **$IO/\overline{M}$ = 0** மற்றும் தலைகீழாக்கி, **E_3=1** கொடுத்துக் குறியீட்டுப் பெயர்ப்பியைச் செயலியக்க வேண்டும்.

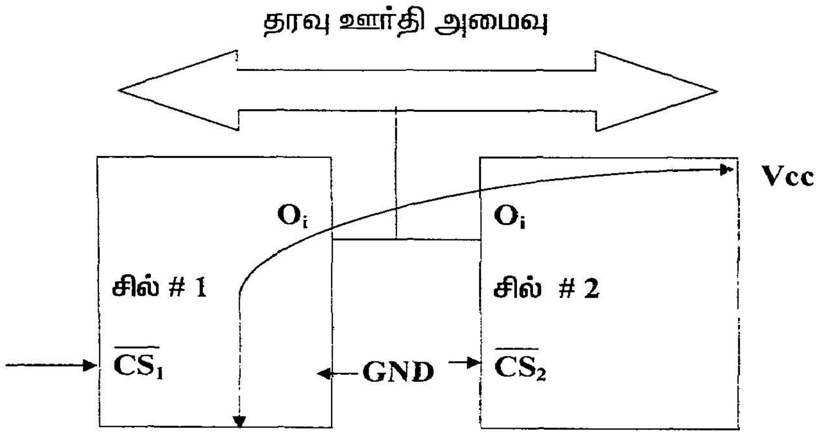
EPROM இன் **$\overline{OE}$** உடன் **$\overline{RD}$** ஐ நேரடியாக இணைக்க வேண்டும்.



படம் **2.22** **2K X 8 EPROM** இடைப்பரிமாறல் (குறியீட்டுப் பெயர்ப்பியில் **IO/M** பயன்படுத்தி)

2.8 ஊர்தி வாதம் மற்றும் 2 வரிக் கட்டுப்பாடு :

இரண்டு நினைவகச் சில்கள், அவற்றின் தரவு வெளியீடுகளுடன், ஒரு பொதுவான தரவு ஊர்தியுடன், கம்பி-**OR** இணைப்பைப் பயன்படுத்திக் கட்டப்பட்டுள்ளன. படம் **2.23**, இதனைக் காட்டுகிறது.



படம் 2.23இல் ஊர்தி வாதத்தின் முக்கியமான தனி வரிக் கட்டுப்பாடு

கீழே கொடுக்கப்பட்டுள்ள படிப்படியான நிகழ்வுகளை, இப்போது ஆலோசிப்போம்.

1. படி செயற்பாடிற்காக $\overline{CS_1}$தாழ்நிலைக்குச் செல்லும்போது, சில் 1 தேர்வு செய்யப்படும்.

2. தாமதத்திற்குப் பிறகு, அதாவது t_{CO}க்குப் பிறகு, தரவு வெளி வரிகள், உயர்நடை தடையிலிருந்து இயல்புநிலைக்கு மாறும்.

3. நுண்முறைப்படுத்தியிலிருந்து தரவைப் படித்தவுடன், $\overline{CS_1}$ உயர்நிலைக்குச் செல்லும். ஆனபோதிலும், முகவரிகள் சில் இரண்டின் $\overline{CS}$ஐ தாழ்நிலைக்கு மாற்றும்.

4. சிறிது தாமதத்திற்குப் பிறகு, இதனை tdf என அழைக்கலாம். சில் ஒன்றின் தரவு வெளியீடுகள், மீண்டும் மிதக்கும் நிலையில் உட்புகும்.

5. $\overline{CS_2}$ தாழ்நிலைக்குச் செல்லுவதனால் tco காலதாமதத்திற்குப் பின்னர் சில் 2இன் தரவு வெளியீடுகள் இயக்கமடையும்.

மேலே கூறிய படிப்படியான நிகழ்வுகளிலிருந்து, t_{df} வை விட tco அதிகமாயிருந்தால், இரண்டு சில்களும் தரவு ஊர்தியைத் தூண்டும் என்பதை அறிய முடியும். அச்சமயத்தில் சில் ஒன்று தேர்வு செய்யப்படாதது, ஊர்தி வாதத்திற்கு முக்கியமானது. சில் 1ஐ படிக்கும்போதும், சில் ஒன்று ஊர்தியில் 1 ஐ அனுப்பும்போதும், சில் இரண்டிலிருந்து சில் ஒன்றுக்குப் புவியின் வழியாகத் தாழ்நடை நடைபெறும்.

இவ்வாறாகப் பாயும் மின்னோட்டமானது, ஒன்று அல்லது அதற்கு மேற்பட்ட விரும்பத்தகாத நிகழ்வுகளை ஏற்படுத்தும். அவையாவன,

1. ஒன்று அல்லது இரண்டு சில்களையும் பழுதடையச் செய்யும்.

2. அதிக மின்னோட்டம், அமைப்பில் துளைபோடும். கொடுக்கப்பட்ட மின்னழுத்தம் Vcc தரவு இழப்பிற்கு இட்டுச் செல்லும்.

நிறைய சில்கள் பயன்படுத்தப்படுகின்ற இரண்டு வரிக் கட்டுப்பாட்டுக் கட்டுமானம், மேலே கூறிய பிரச்சனைகளைத் தெளிவுபடுத்தும். இரண்டாவது கட்டுப்பாட்டு உள்ளீட்டிற்கு, வெளியீடு செயலியக்கம் அல்லது செயல் முடக்கம் என்று பெயர். இப்போது செயலியக்கச் சில்லைப் பயன்படுத்தி, அது சில்லை அணியநிலையிலிருந்து இயக்கநிலைக்கு நகர்த்தும். தரவானது படிப்பிக்கப்பட்டது என்றாலும், உள்ளீடு சில்லிலுள்ள வெளியீட்டுத் தாங்கிகள், தங்களுடைய உயர் நடைதடை நிலையிலிருக்கும். வெளியீடு செயலியக்கம் பெற வேண்டும் எனில், சில்லின் தரவு வெளியீட்டு முனைகளில் மட்டுமே, தரவானது செயலியக்கம் அடையும். வெளியீடு செயல் முடக்கத்திற்குப் பிறகு, வெளியீடு தாங்கிகள், மீண்டும் உயர் நடைதடை நிலைக்குள் நுழையும்.

பயிற்சிகள்

1. 8085 ஆல் எவ்வாறு INTR குறுக்கீடு கையாளப்படுகிறது என்பதை விளக்குக.
2. SIM மற்றும் RIM கட்டளைகளை விளக்குக.
3. மறைக்கப்பட்ட மற்றும் மறைக்கப்படாத குறுக்கீடுகளுக்கு இடையேயான வேறுபாட்டைக் குறிப்பிடுக.
4. 8085இன் குறுக்கீட்டு அமைப்பை விவாதி.
5. வன்பொருள் குறுக்கீடு மற்றும் மென்பொருள் குறுக்கீடு, இவற்றிற்கிடையே உள்ள வேறுபட்டை விளக்குக
6. 8085இல் உள்ள பல்வேறு குறுக்கீடுகளை வரிசைப்படுத்தி, அவற்றின் முந்துரிமைகளைப் பற்றி எழுதுக.
7. 8085இல் TRAP இன் பயன் என்ன ?
8. IN மற்றும் OUT கட்டளைகளை விளக்குக.
9. நேரடி I/O முறையில், எத்தனை உள்ளீடு மற்றும் வெளியீடு துறைமுகங்கள் இணைக்கப்பட்டுள்ளன. காரணங்களை எழுதுக.
10. IN கட்டளைக்கான நேர வரைபடம் வரைந்து, அதனை விளக்குக.
11. உள்ளீடு துறைமுகத்தின் முகவரி FEH உடன் குறியீட்டுப் பெயர்ப்பியைப் பயன்படுத்துக.
12. OUT கட்டளைக்கான நேர வரைபடம் வரைந்து விளக்குக.
13. I/O மற்றும் நினைவக I/O இவற்றிற்கிடையே உள்ள வேறுபாட்டை எழுதுக.

3
இடைப் பரிமாற்றுப் புறவியல் சாதனங்கள் (Interfacing Peripheral Devices)

3.1 முன்னுரை

வெளி உலகத்துடன் தொடர்புகொள்ள நுண்கணினிகள் புறவியலான சாதனங்களைப் பயன்படுத்துகின்றன. பொதுவாகப் பயன்படுத்தக்கூடிய புறவியலான சாதனங்கள், எண்ணிலக்க ஒப்புமை மாற்றி (D/A converter), ஒப்புமை – எண்ணிலக்க மாற்றி (A/D Converter), அச்சுப்பொறி (Printer), வன்வட்டு (hard disk), நெகிழ்வட்டு (Floppy disk), காந்த நாடா (Magnetic tape) போன்றவை. புறவியலான கருவிகள், மின்சுற்றுகள் மூலம் நுண்கணினியோடு இணைக்கப்படுகின்றன. இந்த மின்சுற்றுகள், இடைப் பரிமாற்றுச் சுற்றுகள் எனப்படுகின்றன. இந்தச் சுற்றுகள், உள்ளீடு சாதனங்களிலிருந்து கிடைக்கக்கூடிய தரவுகளை, நுண்கணினிக்கு ஏற்றவாறு மாற்றுகின்றன. வெளியீடு சாதனங்களுடன் தொடர்புடைய இடைப்பரிமாற்றுச் சுற்றுகள், நுண்கணினியிலிருந்து கிடைக்கக்கூடிய வெளியீடுகளைப் புறவியலான கருவிகளுக்கு ஏற்றவாறு மாற்றி அமைக்கின்றன. நுண்முறைப்படுத்தி உற்பத்தியாளர்கள் பொது நோக்க மற்றும் சிறப்பு நோக்க இடைப் பரிமாற்று ச் சில்லுகள, அ திக அளவில் உற்பத்தி செய்கின்றார்கள். சில பொது நோக்க இடைப் பரிமாற்றுச் சாதனங்கள், கீழே கொடுக்கப்பட்டுள்ளன.

i) உள்ளீடு / வெளியீடு நிலையம் (I/O Port).
ii) திட்ட நிரலாக்குப் புறவியல் இடைப் பரிமாற்றுச் சுற்று (Programmable peripheral Interface)
iii) நேரடி நினைவகம் அணுகும் கட்டுப்பாட்டகம் (Direct Memory Access Controller- DMA Controller)
iv) தடங்கல் கட்டுப்பாட்டகம் (Interrupt Controller)
v) தொடர்பு இடைப் பரிமாற்றுச் சுற்று (Communication Interface)
vi) திட்ட நிரலாக்கு எண்ணி / இடைவெளி நேரக்காட்டி (Programmable Counter / Interval Timer)

3.2 திட்ட நிரலாக்குப் புறவியல் இடைப் பரிமாற்றுச் சுற்று (Programmable peripheral Interface)

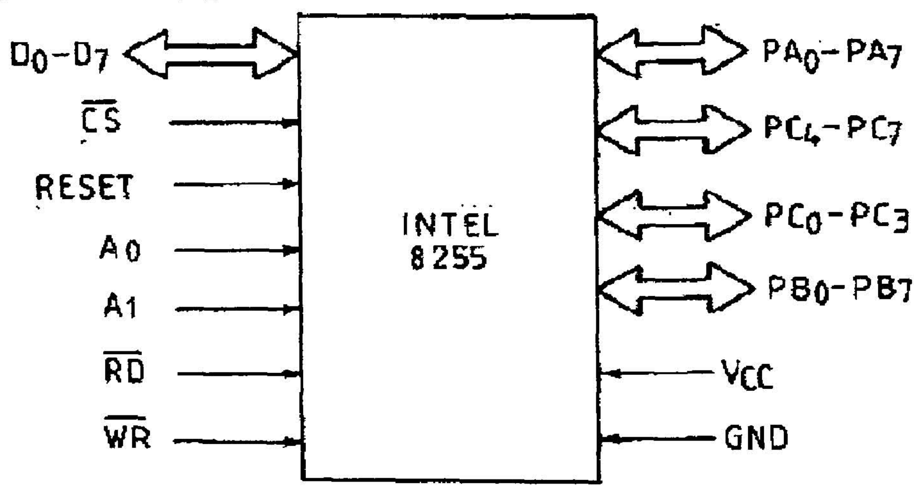
இது ஒரு பல் நிலையச் சாதனமாகும். இந்த நிலையங்களை திட்ட நிரலருக்குத் தேவைப்படும், அளவு பல வழிகளில் திட்ட நிரலாக்கம் செய்து கொள்ளலாம்.

இன்டெல் 8255

இது ஒரு திட்ட நிரலாக்குப் புறவியல் இடைப் பரிமாற்றுச் சாதனம். புறவியலான சாதனங்களை நுண்கணினியோடு இணைப்பதே இதன் முக்கியமான வேலையாகும். இதில் மூன்று 8–பிட் நிலையங்கள் உள்ளன. அவை நிலையம் A, நிலையம் B மற்றும் நிலையம் C. நிலையம் C மேலும் இரண்டு 4-bit நிலையங்களாகப் பிரிக்கப்படுகின்றன. அவை, மேல் நிலையம் C மற்றும் கீழ் நிலையம் C. ஆகவே, மொத்தம் நான்கு நிலையங்கள், இ ன்டெல் 8255 இல் உள்ளன. ஒவ்வொரு நிலையத்தையும் உள்ளீடு நிலையமாகவோ அல்லது வெளியீடு நிலையமாகவோ, திட்ட நிரலாக்கம் செய்து கொள்ளலாம்.

இன்டெல் 8255 இன் கட்டமைப்பு

படம் (3.1)இல் இன்டெல் 8255 இன் மின்முனை அமைப்பு காட்டப்பட்டுள்ளது. இது 40 மின்முனைகளைக் கொண்ட தொகுப்புச் சுற்றுச் சில்லாகும். இது இயங்குவதற்கு, +5V மின்னழுத்தம் தேவைப்படுகிறது.



படம் (3.1) இன்டெல் 8255இன் முனை அமைப்பு

வெவ்வேறு நிலையங்களுக்கான மின்முனை அமைப்பு

$PA_0 - PA_7 \rightarrow$ நிலையம் A இன் **8** மின்முனைகள்

$PB_0 - PB_7 \rightarrow$ நிலையம் B இன் **8** மின்முனைகள்

$PC_0 - PC_3 \rightarrow$ நிலையம் $C_{\text{கீழ்}}$ இன் **4** மின்முனைகள்

$PC_4 - PC_7 \rightarrow$ நிலையம் $C_{\text{மேல்}}$ இன் **4** மின்முனைகள்

முக்கியமான கட்டுப்பாட்டுச் சைகைகள்

$\overline{CS}$: இது ஒரு சில்லைத் தேர்ந்தெடுக்கும் சைகையாகும். இது தாழ்நிலையில் இருந்தால், மையச்செயலகத்திற்கும் மற்றும் **8255**க்கும் இடையே தொடர்பை ஏற்படுத்தும்.

$\overline{RD}$: இந்தச் சைகை தாழ்நிலையில் இருக்கும்போது, **8255** அலகானது தரவு அல்லது நிலைப்பாடு தகவல்களை, மையச் செயலகத்திற்கு அனுப்பும்.

$\overline{WR}$: இந்தச் சைகை தாழ்நிலையில் இருந்தால், மையச் செயலகம் தரவுகளை **8255**க்கு அனுப்பும்.

A_0 மற்றும் A_1: உள்ளீடு நிலையம் மற்றும் கட்டுப்பாட்டு வார்த்தைப் பதிவேட்டைத் தேர்ந்தெடுக்கும் வேலையை A_0 மற்றும் A_1 சைகைகள் செய்கின்றன. இரண்டு **8255** அலகுகள் பயன்படுத்தப்பட்டால், உள்ளீடு நிலையங்களுக்கான முகவரி, கீழ்கண்டவாறு இருக்கும்.

8255இன் முதல் அலகு அதாவது 8255.1

நிலையம் / கட்டுப்பாட்டு வார்த்தைப் பதிவேடு	முகவரி
நிலையம் A	**00**
நிலையம் B	**01**
நிலையம் C	**02**
கட்டுப்பாட்டு வார்த்தைப் பதிவேடு	**03**

8255இன் இரண்டாம் அலகு அதாவது 8255.2

நிலையம் / கட்டுப்பாட்டு வார்த்தைப் பதிவேடு	முகவரி
நிலையம் A	**08**
நிலையம் B	**09**
நிலையம் C	0A
கட்டுப்பாட்டு வார்த்தைப் பதிவேடு	0B

8255 அலகின் இயங்கும் முறைகள்

இன்டெல் 8255க்கு, மூன்று இயங்கும் முறைகள் உள்ளன. அவை

முறை 0 – உள்ளீடு அல்லது வெளியீடு நிலையமாக இயங்கச் செய்தல்.

முறை 1 – விரைவான இடைவெளியில் உள்ளீடு அல்லது வெளியீடு நிலையமாக இயங்கச் செய்தல்

முறை 2 – இருதிசை நிலையமாக இயங்கச் செய்தல்

முறை 0 : இந்த முறையில், இன்டெல் 8255 இன் நான்கு நிலையங்களில், ஒவ்வொரு நிலையத்தையும், உள்ளீடு நிலையமாகவோ அல்லது வெளியீடு நிலையமாகவோ இயங்கச் செய்யலாம்.

முறை 1 – இந்த முறையில், ஒரு நிலையத்தை விரைவான இடைவெளியில் உள்ளீடு அல்லது வெளியீடு நிலையமாக இயங்கச் செய்யலாம். நிலையம் A மற்றும் B ஆகியவை இந்த முறையில் இயக்கப்படுகின்றன. நிலையம் Cயின் ஆறு முனைகள், A மற்றும் B நிலையங்களைக் கட்டுப்படுத்தப் பயன்படுகின்றன. நிலையம் Bஐ உள்ளீடு அல்லது வெளியீடு நிலையமாகச் செயல்படுத்தும்போது, PC_0, PC_1 மற்றும் PC_2 ஆகியவை Bஐ கட்டுப்படுத்த உதவுகின்றன. நிலையம் A உள்ளீடு நிலையமாகச் செயல்படும்போது, PC_3, PC_4 மற்றும் PC_5 ஆகியவை நிலையம்Aஐ கட்டுப்படுத்த உதவுகின்றன. நிலையம் Cஇன் மற்ற முனைகளான PC_6 மற்றும் PC_7 ஆகியவற்றை, உள்ளீடாகவோ அல்லது வெளியீடாகவோ பயன்படுத்தலாம். நிலையம் A வெளியீடு நிலையமாகச் செயல்படும்போது, PC_3 மற்றும் PC_6 ஆகியவை, Aஐ கட்டுப்படுத்த பயன்படுகிறது. PC_4 மற்றும் PC_5 ஆகியவற்றை, உள்ளீடாகவோ அல்லது வெளியீடாகவோ பயன்படுத்தலாம். நிலையம் A முறை 1இல் இயங்கினால், நிலையம் B முறை 0இல் இயங்கும்.

முறை 2 : இந்த முறையில், நிலையத்தை விரைவான இடைவெளியில் இருதிசையில் இயங்கச் செய்யலாம். நிலையம் Aஐ, இந்த முறையில் இயங்கச் செய்யலாம். PC_3, PC_4, PC_5, PC_6, PC_7 ஆகியவை, Aஐ

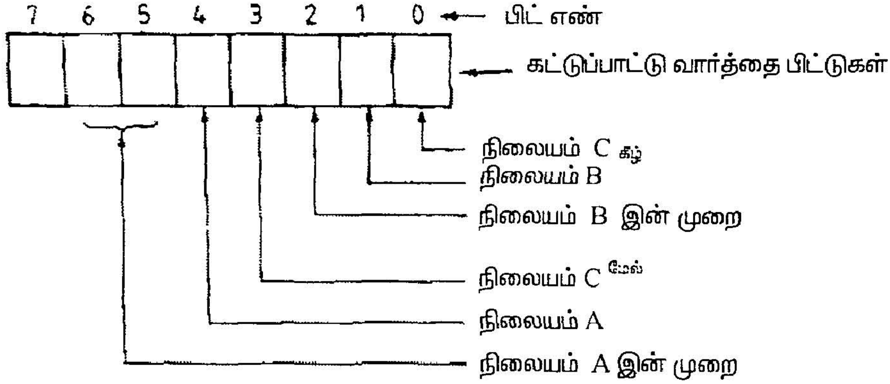
கட்டுப்படுத்தப் பயன்படுகின்றன. நிலையம் Aஐ முறை 2இல் பயன்படுத்தினால், Bஐ முறை 1இல் அல்லது 0இல் பயன்படுத்தலாம்.

கட்டுப்பாட்டுத் தொகுப்பு (Control groups)

உள்ளீடு, வெளியீடு நிலையங்களின் 24 கம்பிகள், கட்டுப்பாட்டு வேலைக்காக, இரண்டு தொகுப்புகளாகப் பிரிக்கப்படுகின்றன. அவை தொகுப்பு A மற்றும் தொகுப்பு B ஆகும். நிலையம் A மற்றும் நிலையம் $C_{மேல்}$ ஆகியவை, தொகுப்பு Aஇல் இருக்கும். நிலையம் B மற்றும் நிலையம் $C_{கீழ்}$ ஆகியவை, தொகுப்பு Bஇல் இருக்கும்.

கட்டுப்பாட்டு வார்த்தை (Control word)

தேவைகளுக்கேற்ப, ஒரு நிலையத்தை உள்ளீடு நிலையமாகவோ அல்லது வெளியீடு நிலையமாகவோ, திட்ட நிரலாக்கம் செய்து கொள்ளலாம். நிலையங்களைத் திட்ட நிரலாக்கம் செய்வதற்குக் கட்டுப்பாட்டு வார்த்தை உருவாக்கப்படுகிறது. நிலையங்களின் வேலை மற்றும் அவை இயங்கும் முறைகள் போன்ற தகவல்கள், கட்டுப்பாட்டு வார்த்தையில் அடங்கி இருக்கும். மையச் செயலகம், இந்தக் கட்டுப்பாட்டு வார்த்தையை, 8255 அலகுக்கு அனுப்பும். கட்டுப்பாட்டு வார்த்தையின் பிட்டுகள், படம் (3.2)இல் காட்டப்பட்டுள்ளன. 8255 அலகிலுள்ள கட்டுப்பாட்டு வார்த்தை, பதிவேட்டில் எழுதப்படும். ஒரு நிலையம் உள்ளீடு நிலையமாகச் செயல்படவேண்டுமா அல்லது வெளியீடு நிலையமாகச் செயல்படவேண்டுமா என்ற வரையறுப்புக்குத் தகுந்தவாறு, அந்த நிலையத்தின் கட்டுப்பாட்டு வார்த்தை பிட்டு, ஒன்றாகவோ அல்லது சுழியாகவோ அமைவு பெறும். ஒரு நிலையம் உள்ளீடு நிலையமாகச் செயல்பட வேண்டுமென்றால், அந்த நிலையத்தைச் சேர்ந்த பிட், ஒன்று 1 என்று அமைவு பெறும். வெளியீடு நிலையமாகச் செயல்பட வேண்டுமென்றால், அந்த நிலையத்தைச் சேர்ந்த பிட், சுழி என்று அமைவு பெறும். கட்டுப்பாட்டு வார்த்தை பிட்டுகளின் விரிவான விளக்கம், கீழே கொடுக்கப்பட்டுள்ளது.



படம் – 3.2

இன்டெல் 8255இன் கட்டுப்பாட்டு வார்த்தை பிட்டுகள்

பிட் எண் 0 → இது நிலையம் $C_{கீழ்}$ இன் பிட்டாகும்.

பிட் எண் 1 → இது நிலையம் B இன் பிட்டாகும்.

பிட் எண் 2 → இந்த பிட் நிலையம் B தேர்ந்தெடுக்கும் முறைக்கானது. நிலையம் B முறை O இல் செயல்பட வேண்டுமென்றால், பிட்டானது சுழி என்று அமைவு பெறும். நிலையம் B முறை 1இல் செயல்பட வேண்டுமென்றால், பிட்டானது ஒன்று என்று அமைவு பெறும்.

பிட் எண் 3 → இது நிலையம் $C_{மேல்}$ இன் பிட்டாகும்.

பிட் எண் 4 → இது நிலையம் A இன் பிட்டாகும்.

பிட் எண் 5 மற்றும் 6 → இந்த பிட்டுகள், நிலையம் A இயங்கும் முறையை வரையறுக்கின்றன. நிலையம் A இயங்கும் முறைக்குத் தகுந்தவாறு, பிட் எண் 5 மற்றும் 6, பின்வருமாறு வரையறுக்கப்படுகிறது.

நிலையம் Aஇன் முறைகள்	பிட் எண் 6	பிட் எண் 5
முறை 0	0	0
முறை 1	0	1
முறை 2	1	0

பிட் எண் 7	→ நிலையங்கள் A, B மற்றும் C உள்ளீடு நிலையமாகவோ அல்லது வெளியீடு நிலையமாகவோ செயல்பட்டால், இந்த பிட்டானது ஒன்று என்று அமைவு பெறும். நிலையம் Cஇன் முனைகள் அமைவு பெற்றாலோ அல்லது மறு அமைவு பெற்றாலோ, இந்த பிட் 'சுழி' என்று அமைவு பெறும்.

முறை 0 இயக்கத்திற்கான 8255 அலகின் நிலையங்களின் வெவ்வேறு கட்டமைப்பிற்குத் தகுந்த கட்டுப்பாட்டு வார்த்தை, அட்டவணை **(3.1)** இல் கொடுக்கப்பட்டுள்ளது.

அட்டவணை 3.1. முறை O இயக்கத்திற்கான 8255 அலகின் கட்டுப்பாட்டு வார்த்தைகள்

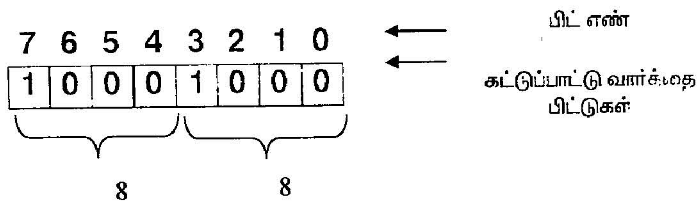
கட்டுப்பாட்டு வார்த்தை பிட்டுகள்								கட்டுப்பாட்டு வார்த்தை	நிலையம் A	நிலையம் C மேல்	நிலையம் B	நிலையம் C கீழ்
7	6	5	4	3	2	1	0					
1	0	0	0	0	0	0	0	80	வெளியீடு	வெளியீடு	வெளியீடு	வெளியீடு
1	0	0	0	0	0	0	1	81	வெளியீடு	வெளியீடு	வெளியீடு	உள்ளீடு
1	0	0	0	0	0	1	0	82	வெளியீடு	வெளியீடு	உள்ளீடு	வெளியீடு
1	0	0	0	0	0	1	1	83	வெளியீடு	வெளியீடு	உள்ளீடு	உள்ளீடு
1	0	0	0	1	0	0	0	88	வெளியீடு	உள்ளீடு	வெளியீடு	வெளியீடு
1	0	0	0	1	0	0	1	89	வெளியீடு	உள்ளீடு	வெளியீடு	உள்ளீடு
1	0	0	0	1	0	1	0	8A	வெளியீடு	உள்ளீடு	உள்ளீடு	வெளியீடு
1	0	0	0	1	0	1	1	8B	வெளியீடு	உள்ளீடு	உள்ளீடு	உள்ளீடு
1	0	0	1	0	0	0	0	90	உள்ளீடு	வெளியீடு	வெளியீடு	வெளியீடு
1	0	0	1	0	0	0	1	91	உள்ளீடு	வெளியீடு	வெளியீடு	உள்ளீடு
1	0	0	1	0	0	1	0	92	உள்ளீடு	வெளியீடு	உள்ளீடு	வெளியீடு
1	0	0	1	0	0	1	1	93	உள்ளீடு	வெளியீடு	உள்ளீடு	உள்ளீடு
1	0	0	1	1	0	0	0	98	உள்ளீடு	உள்ளீடு	வெளியீடு	வெளியீடு
1	0	0	1	1	0	0	1	99	உள்ளீடு	உள்ளீடு	வெளியீடு	உள்ளீடு
1	0	0	1	1	0	1	0	9A	உள்ளீடு	உள்ளீடு	உள்ளீடு	வெளியீடு
1	0	0	1	1	0	1	1	9B	உள்ளீடு	உள்ளீடு	உள்ளீடு	உள்ளீடு

உதாரணம்

முறை 0 இயக்கத்திற்கான இன்டெல் 8255 நிலையங்களின் கீழே கொடுக்கப்பட்ட வெவ்வேறு கட்டமைப்பிற்குத் தகுந்த கட்டுப்பாட்டு வார்த்தையை உருவாக்குக.

நிலையம் A	-	வெளியீடு
நிலையம் B	-	வெளியீடு
நிலையம் $C_{\text{கீழ்}}$	-	வெளியீடு
நிலையம் $C_{\text{மேல்}}$	-	உள்ளீடு

இந்தக் கட்டமைப்பிற்கான கட்டுப்பாட்டு வார்த்தை பிட், படம் (3.3)இல் காட்டப்பட்டுள்ளது. கட்டுப்பாட்டு வார்த்தை 88H ஆகும்.



படம் – 3.3

3.3 திட்ட நிரலாக்கு நேரடி நினைவகம் அணுகும் கட்டுப்பாட்டகம் (Programmable Direct Memory Access controller or DMA controller)

முன்னுரை

உள்ளீடு/வெளியீடு சாதனங்களிலிருந்து நினைவகத்திற்குத் தரவுகளைச் சேமக்கலம் வழியாக மாற்றம் செய்வதற்கும், அல்லது நினைவகத்திலிருந்து உள்ளீடு/வெளியீடு சாதனங்களுக்குச் சேமக்கலம் வழியாகத் தரவுகளை மாற்றம் செய்வதற்கும் அதிக நேரம் செலவிட வேண்டும். இந்தக் குறையைச் சரிசெய்ய நேரடி நினைவகம் அணுகும் (Direct Memory Access or DMA) யுக்தி கையாளப்படுகிறது. இந்த யுக்தி மூலம் தரவுகளை நேரடியாக மாற்றம் செய்யலாம். இந்த யுக்தியைக் கையாளும்போது, தரவு மற்றும் முகவரி ஊர்திகள், புறவியலான சாதனங்களின் கட்டுப்பாட்டில் இருக்கும். DMA முறைப்படி தரவுகளை மாற்றம் செய்வதற்கு, நினைவக முகவரி, மற்றும் பைட் எண்ணிக்கையைச் சேமித்து வைக்க, உள்ளீடு/வெளியீடு சாதனங்களுக்கென்று தனிப் பதிவேடுகள் இருக்க வேண்டும். மேலும், தரவுகளை மாற்றம் செய்யத் தேவையான கட்டுப்பாட்டுச் சைகைகளை, உள்ளீடு/வெளியீடு சாதனங்கள்

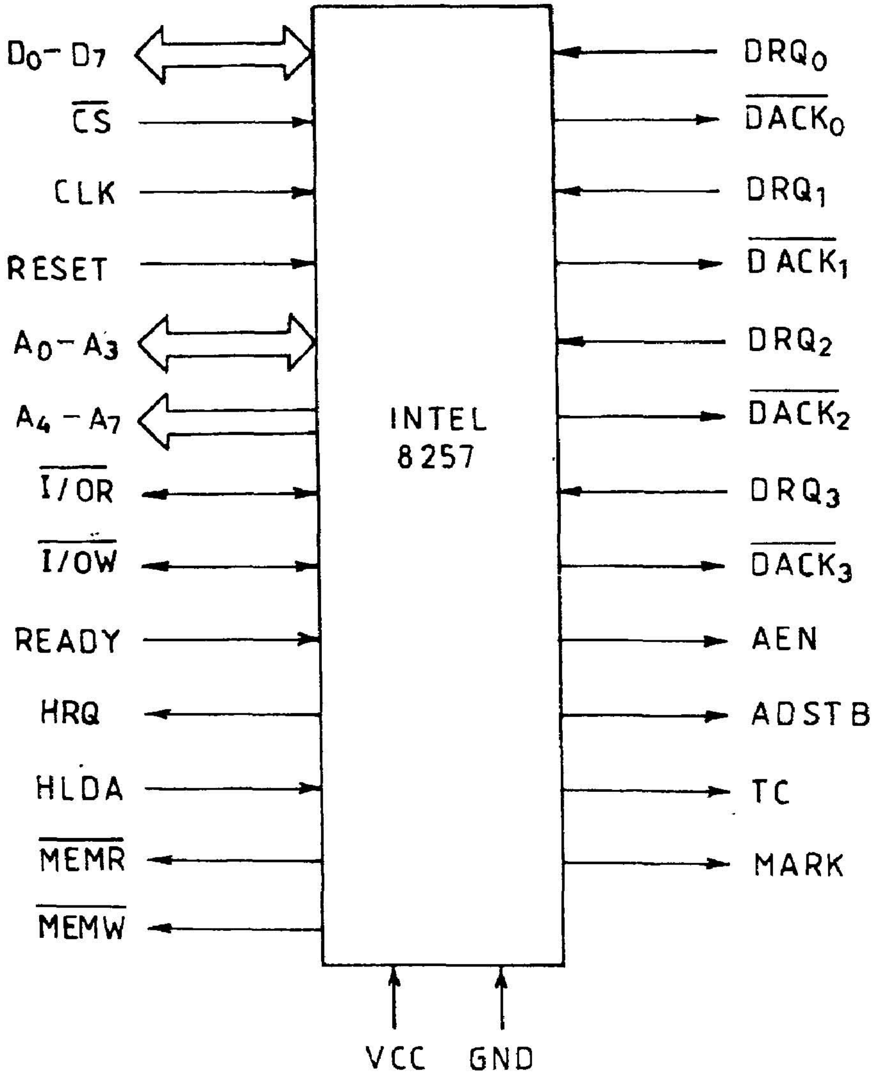
உற்பத்தி செய்ய வேண்டும். ஆனால், இந்த வசதிகள், உள்ளீடு/வெளியீடு சாதனங்களில் கிடையாது. எனவே உள்ளீடு/வெளியீடு சாதனங்களை நுண்முறைப்படுத்தியோடு இடைப் பரிமாற்றம் செய்ய, உற்பத்தியாளர்கள் இன்டெல் 8257 மற்றும் இன்டெல் 8237 போன்ற திட்ட நிரலாக்கு நேரடி நினைவகம் அணுகும் கட்டுப்பாட்டகத்தை உற்பத்தி செய்தனர். DMA முறைப்படி தரவுகளை மாற்றம் செய்யத் தேவையான அத்தனை வசதிகளும், இன்டெல் 8257இல் உள்ளது. அவற்றின் விரிவான விளக்கத்தைப் பின்வரும் பிரிவுகளில் பார்ப்போம்.

இன்டெல் 8257

இந்தச் சில்லின் முனை அமைப்பு, படம் (3.4)இல் காட்டப்பட்டுள்ளது. இது நான்கு தடங்களைக் (Channel) கொண்ட கட்டுப்பாட்டகமாகும். இந்தத் தொகுப்புச் சில்லில் 40 முனைகள் உள்ளன. இது இயங்குவதற்கு +5V மின்னழுத்தம் தேவைப்படுகிறது. இந்த அலகின் மூலம், நான்கு உள்ளீடு/வெளியீடு சாதனங்களை நுண்முறைப்படுத்தியோடு இடைப் பரிமாற்றம் செய்யலாம். படித்தல், எழுதுதல் மற்றும் சரிபார்த்தல் வேலைகளை, இந்த அலகால் செய்ய முடியும். படித்தல் இயக்கத்தைச் செயல்படுத்தும்போது, தரவுகளை நினைவகத்திலிருந்து உள்ளீடு/வெளியீடு சாதனங்களுக்கு மாற்றம் செய்ய முடியும். எழுதுதல் இயக்கத்தைச் செயல்படுத்தும்போது, தரவுகள் உள்ளீடு வெளியீடு சாதனங்களிலிருந்து நினைவகத்திற்கு மாற்றம் செய்யப்படும். ஒவ்வொரு தடத்திலும், இரண்டு 16–பிட் பதிவேடுகள் இருக்கும். அவை DMA முகவரிப் பதிவேடு மற்றும் பைட் எண்ணிக்கைப் பதிவேடுகளாகும். DMA முகவரிப் பதிவேட்டில், முதலில் அணுக வேண்டிய நினைவக இருப்பிடத்தின் முகவரி சேமிக்கப்படும். அடுத்த DMA சுற்றில், இந்தப் பதிவேட்டில், அடுத்ததாக அணுக வேண்டிய நினைவக இருப்பிடத்தின் முகவரி சேமிக்கப்படும். பைட் எண்ணிக்கைப் பதிவேட்டின் 14 சிறும முக்கியத்துவம் வாய்ந்த பைட்டுகளில், உள்ளீடு/வெளியீடு சாதனங்களிலிருந்து நினைவகத்திற்கு மாற்றம் செய்யக்கூடிய பைட்டுகள் சேமிக்கப்படும். அதாவது 2^{14} பைட் தரவுகள், உள்ளீடு/வெளியீடு சாதனங்களிலிருந்து நினைவகத்திற்கு மாற்றம் செய்யப்படும் அல்லது நினைவகத்திலிருந்து உள்ளீடு/வெளியீடு சாதனங்களுக்கு மாற்றம் செய்யப்படும். மீதமுள்ள 2 பெரும முக்கியத்துவம் வாய்ந்த பிட்டுகள், கட்டுப்பாட்டகம், தடங்களில் செய்யக்கூடிய வேலையை உணர்த்தும். இந்தப் பதிவேடுகளைத்

தவிர 8257 அலகில் முறை அமைவுப் பதிவேடு (Mode Set Register) மற்றும் நிலைப்பாடு பதிவேடுகள் உள்ளன.

இன்டெல் 8257 அலகின் முக்கியமான முனைகளைப் பற்றிக் காண்போம்.



படம் (3.4) இன்டெல் 8257இன் முனைஅமைப்பு

DRQ_0 – DRQ_3 : இவை DMA வேண்டுதல் கோடுகளாகும். ஒரு உள்ளீடு/வெளியீடு சாதனம், DMA வேண்டுதலை இந்தக் கோட்டின் வழியாக அனுப்பும்.

$\overline{DACK_0}$ – $\overline{DACK_3}$: DMA வேண்டுதலைப் பெற்றுக் கொண்டதற்கு அடையாளமாக, 8257 அலகு, ஒரு சைகையை, இந்தக் கோடுகளில் ஏதாவது ஒன்றின் வழியாக உள்ளீடு/வெளியீடு சாதனத்திற்குத் திரும்ப அனுப்பும். இதன் மூலம், அந்தச் சாதனம் DMA தரவு மாற்றத்திற்குத் தேர்ந்தெடுக்கப்பட்டுள்ளது என்பதைத் தெரிவிக்கும்.

A_0 – A_7 : இவை முகவரி கோடுகளாகும்.

A_0 – A_3 : கோடுகள் இருதிசையில் முகவரியை எடுத்துச் செல்லும். 8257 அலகு உற்பத்தி செய்யக்கூடிய 16–பிட் நினைவக முகவரியில், நான்கு சிறும முக்கியத்துவம் வாய்ந்த பிட்டுகளை, இந்தக் கோடுகள் எடுத்துச் செல்லும். நான்கிலிருந்து ஏழாவது பிட்டுவரை உள்ள சிறும முக்கியத்துவம் வாய்ந்த பிட்டுகளை, A_4 – A_7 கோடுகள், 8257 அலகிலிருந்து நினைவகத்திற்கு எடுத்துச் செல்லும்.

D_0 – D_7 : இவை தரவுக் கோடுகளாகும். DMA சுற்றின் தொடக்கத்தில், 8257 அலகிலிருந்து 8 பெரும முக்கியத்துவம் வாய்ந்த பிட்டுகளை எடுத்துச் செல்லும். இந்தப் பிட்டுகளை 8212 அலகில் வைத்து விட்டு, DMA சுற்றின் கடைசியில் தரவுகளைக் கையாளப் பயன்படுத்தப்படும்.

AEN : முகவரிச் செயலியக்கச் சைகையாகும்.

ADSTB : இந்தச் சைகை உயர் நிலையில் இருக்கும் போது, தரவு ஊர்தியில் அனுப்பப்பட்ட 8 பெரும முக்கியத்துவம் வாய்ந்த முகவரி பிட்டுகள், 8212 அலகில் வைக்கப்படும்.

$\overline{CS}$: சில்லினைத் தேர்வு செய்யும் சைகையாகும்.

$\overline{I/OR}$: இது இரு திசைகளில் செயல்படக்கூடிய உள்ளீடு/வெளியீடு படித்தல் சைகையாகும்.

$\overline{I/OW}$: இது இரு திசைகளில் செயல்படக்கூடிய உள்ளீடு/வெளியீடு எழுதுதல் சைகையாகும்.

$\overline{MEMR}$: நினைவகப் படித்தல் சைகையாகும்.

$\overline{\text{MEMW}}$: நினைவக எழுதுதல் சைகையாகும்.

TC : பைட் எண்ணிக்கையைக் குறிப்பிடும் சைகையாகும்.

HRQ : ஊர்திகளின் பயன்பாட்டை நிறுத்த வேண்டி, **8257** அலகால் மையச் செயலகத்திற்கு அனுப்பப்படும் சைகை.

HLDA: HOLD சைகையைப் பெற்றுக் கொண்டதற்கு அடையாளமாக, மையச் செயலகம் திரும்ப அனுப்பும் சைகை.

CLK : கடிகார உள்ளீடு.

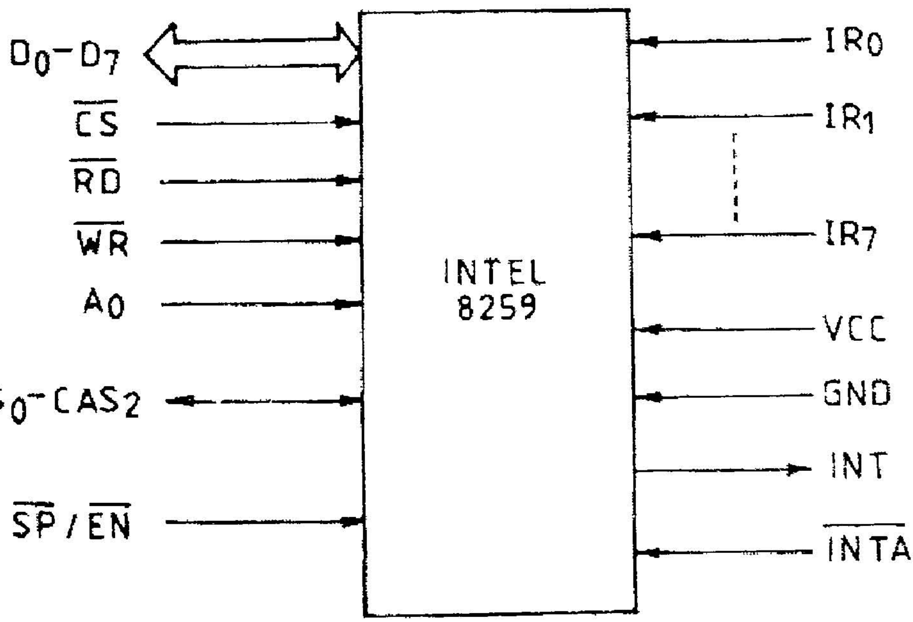
முதலில் ஒரு உள்ளீடு/வெளியீடு சாதனம், DMA தரவு மாற்றத்தை வேண்டி, நான்கு DRQ கோடுகளில், ஒரு கோட்டின் வழியாகச் சைகையை **8257** அலகிற்கு அனுப்பும். இந்தச் சைகையைப் பெற்றுக் கொண்ட பிறகு, **8257** அலகானது, ஊர்திகளைப் பயன்படுத்துவதை நிறுத்த வேண்டி, மையச் செயலகத்திற்கு ஒரு வேண்டுதல் சைகையை HRQ கோட்டின் வழியாக அனுப்பும். இந்தச் சைகையைப் பெற்றுக் கொண்டதற்கு அடையாளமாக, மையச் செயலகம் ஒரு சைகையை HLDA கோட்டின் வழியாகத் திரும்ப **8257** அலகிற்கு அனுப்பும். பிறகு **8257** அலகானது, DMA தரவு மாற்றத்திற்குத் தயார் என்பதை உள்ளீடு/வெளியீடு சாதனத்திற்கு உணர்த்தும் வகையில், ஒரு சைகையை $\overline{\text{DACK}}$ கோட்டின் வழியாக அனுப்பும். நினைவக இருப்பிடத்தின் முகவரி, முகவரி மற்றும் தரவு ஊர்தி வழியாக **8257** அலகிலிருந்து சைகை அனுப்பப்படும்.

DMA படித்தல் சுற்றிற்காக, நினைவகத்திலிருந்து உள்ளீடு வெளியீடு சாதனங்களுக்குத் தரவுகளை மாற்றம் செய்ய **8257** அலகால் $\overline{\text{MEMR}}$ மற்றும் $\overline{\text{I/OW}}$ ஆகிய இரண்டு கட்டுப்பாட்டுச் சைகைகள் வெளியிடப்படும். $\overline{\text{MEMR}}$ சைகை தாழ்நிலையில் இருக்கும் போது, தேர்ந்தெடுக்கப்பட்ட நினைவகத்திலிருந்து உள்ளீடு/வெளியீடு சாதனத்திற்குத் தரவுகள் மாற்றம் செய்யப்படும். $\overline{\text{I/OW}}$ சைகை தாழ்நிலையில் இருக்கும் போது, நினைவகத்திலிருந்து அனுப்பப்பட்ட தரவுகள், உள்ளீடு வெளியீடு சாதனத்தில் எழுதப்படும். இதே போன்று, DMA எழுதுதல் சுற்றில் தேர்ந்தெடுக்கப்பட்ட

நினைவக இருப்பிடம் மற்றும் உள்ளீடு வெளியீடு சாதனங்களுக்கிடையே, தரவு மாற்றம் $\overline{MEMW}$ மற்றும் $\overline{I/OR}$ சைகையால் கட்டுப்படுத்தப்படும். நினைவக இருப்பிடம் மற்றும் உள்ளீடு/வெளியீடு சாதனங்களுக்கிடையே ஒரு பைட் தரவு மாற்றம் செய்யப்பட்ட பிறகு, பைட் எண்ணிக்கை சுழியானால், DMA தரவு மாற்றம் முடிந்து விட்டதாகக் கருதப்படும்.

3.4. திட்டநிரலாக்குத் தடங்கல் கட்டுப்பாட்டகம்–இன்டெல் 8259 (Programmable Interrupt Controller-Intel 8259)

தடங்கல் சைகையைப் பயன்படுத்திப் பல உள்ளீடு/வெளியீடு சாதனங்கள், தரவுகளை மையச் செயலகத்திற்கு அனுப்பும்போது, திட்ட நிரலாக்குத் தடங்கல் கட்டுப்பாட்டகம் பயன்படுத்தப்படுகிறது. இன்டெல் 8259, ஒரு தனி தொகுப்புச் சில்லாகும். இந்தச் சில்லில் 28 மின்முனைகள் உள்ளன. இது இயங்குவதற்கு +5V மின்னழுத்தம் தேவைப்படுகிறது. இன்டெல் 8259இன் முனைஅமைப்பு, படம் (3.5)இல் காட்டப்பட்டுள்ளது. இதன் மின்முனைகளின் விளக்கத்தைப் பற்றிக் காண்போம்.



படம் – (3.5) இன்டெல் 8259இன் முனை அமைப்பு

$\overline{CS}$: இது சில்லைத் தேர்ந்தெடுக்கும் சைகையாகும்.

$\overline{WR}$: இந்தச் சைகை தாழ்நிலையில் இருக்கும்போது, இன்டெல் **8259** அலகானது, மையச் செயலகத்திலிருந்து வரக்கூடிய கட்டளை வார்த்தையை (Command word) ஏற்றுக் கொள்ளும்.

$\overline{RD}$: இந்தச் சைகை தாழ்நிலையில் இருக்கும் போது, இன்டெல் **8259** அலகானது, பல நிலைப்பாடு சைகைகளைத் தரவு ஊர்தி மூலமாக மையச் செயலகத்திற்கு அனுப்பும்.

D_0 – D_7 : இது இருதிசை தரவு ஊர்தியாகும். கட்டுப்பாடு, நிலைப்பாடு மற்றும் தடங்கல் சைகைகள், இந்தத் தரவு ஊர்தி வழியாக அனுப்பப்படுகிறது.

CAS_0 – CAS_2 : ஒன்றிணைக்கப்பட்ட கம்பிகள்

$\overline{EN}$: இந்தச் சைகை தாழ்நிலையில் இருக்கும் போது, இடைநிலை நினைவகம் (Buffer) செயலியக்கப்படுகிறது.

INT : இது தடங்கல் சைகையாகும். இந்தச் சைகை, **8259** அலகிலிருந்து மையச் செயலகத்திற்கு அனுப்பப்படுகிறது.

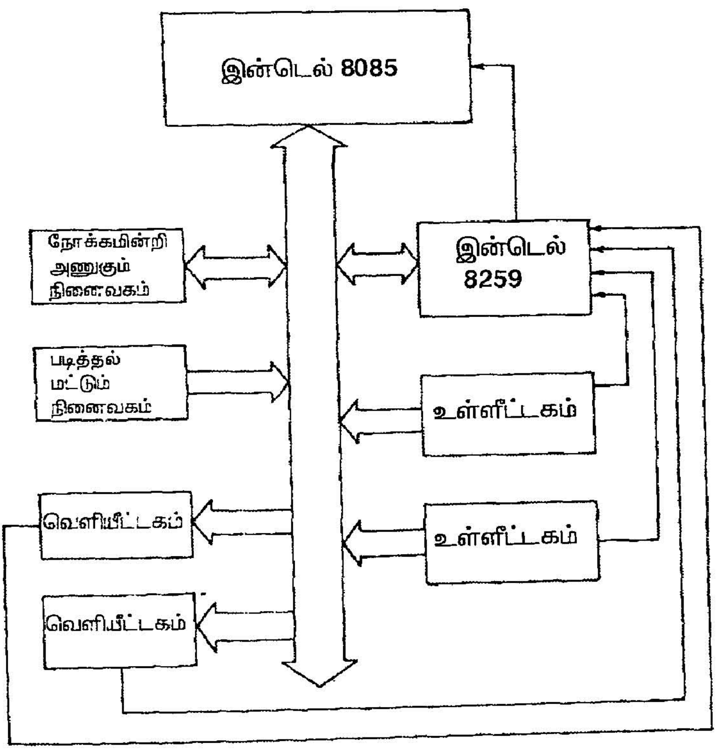
$\overline{INTA}$: தடங்கல் சைகையைப் பெற்றுக் கொண்டதற்கு அடையாளமாக, மையச் செயலகம் திரும்ப அனுப்பும் சைகையாகும்.

IR_0 – IR_7 : இந்தக் கம்பிகளின் வழியாக உள்ளீடு/வெளியீடு சாதனங்கள், தடங்கல் வேண்டுதல் சைகையை **8259** அலகிற்கு அனுப்புகின்றன.

A_0 : இது முகவரிக் கோடாகும். இது $\overline{RD}$, $\overline{WR}$ மற்றும் $\overline{CS}$ ஆகிய சைகைகளுடன் சேர்ந்து செயல்படும்.

திட்ட நிரலாக்குத் தடங்கல் கட்டுப்பாட்டகம் மற்றும் உள்ளீடு/ வெளியீடு சாதனங்கள் எவ்வாறு நுண்கணினியோடு இணைக்கப்படுகின்றன என்பது, படம் **(3.6)**இல் காட்டப்பட்டுள்ளது. எட்டு உள்ளீடு/வெளியீடு சாதனங்களை, IR_0 – IR_7 கம்பிகள் மூலம் **8259** அலகுடன் இணைக்க முடியும். தடங்கல் கட்டுப்பாட்டகம்,

உள்ளீடு/வெளியீடு சாதனங்களிலிருந்து வேண்டுதல்களை ஏற்றுக் கொண்டு, அவற்றில் எதற்கு அதிக முக்கியத்துவம் கொடுக்க வேண்டும் என்பதைத் தீர்*மானிக்கும். பிறகு 8259 அலகு, தடங்கல் சைகையை INT கம்பி மூலம் நுண்முறைப்படுத்திக்கு அனுப்பும். இதை ஏற்றுக் கொண்டதற்கு அடையாளமாக நுண்முறைப்படுத்தி* $\overline{\text{INTA}}$ *என்ற சைகையைத் திரும்ப* அனுப்பும். இந்தச் சைகையை பெற்றுக் கொண்ட பிறகு, குறைந்த முக்கியத்துவம் வாய்ந்த தடங்கல் சைகைகளின் செயல்பாடுகள் தடுக்கப்படும். மற்றும் 8259 அலகு CALL கட்டளையை நுண்முறைப்படுத்திக்கு அனுப்பும். ஒன்றாக இணைக்கப்பட்ட பல 8259 சில்லுகளால், எந்த ஒரு கூடுதல் மின்சுற்றுமின்றி, அறுபத்தி நான்கு தடங்கல் சைகைகளைப் பெற்றுக் கொள்ள முடியும்.

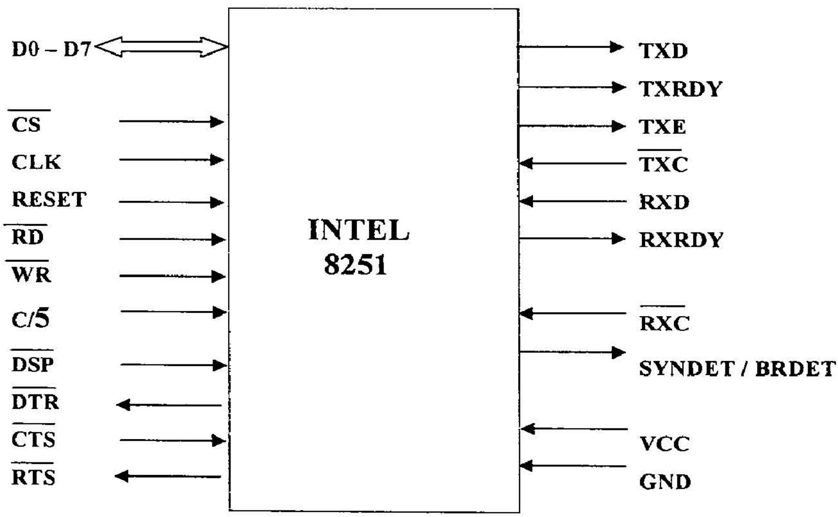


படம் – 3.6

8259 அலகு மற்றும் உள்ளீடு / வெளியீடு சாதனங்களின் இடைப் பரிமாற்றுச் சுற்று

3.5 திட்டநிரல் பரப்பு இடைப்பரிமாறல் (INTEL 8251) :

திட்டநிரல் பரப்பு இடைப்பரிமாறல், தொடர் தரவை அனுப்புவதற்காகப் பயன்படுகிறது. இன்டெல் 8251 என்பது, ஒரு திட்டநிரல் பரப்பு இடைப்பரிமாறல். தொடர் தரவை அனுப்புவதற்கு / பெறுவதற்கு 8251ஐ பயன்படுத்தலாம். நுண்முறைப்படுத்தியிலிருந்து தரவைப் பக்க வடிவ அமைப்பில் ஏற்றுக்கொண்டு, அதனைத் தொடர் தரவாக மாற்றி அனுப்புகிறது. மேலும், அது தொடர் தரவினை ஏற்றுக்கொண்டு, அதனைப் பக்க தரவாக மாற்றி CPUக்குத் தரவைப் பக்க வடிவ அமைப்பில் அனுப்புகிறது. இன்டெல் 8251இன் வரைபடம், படம் 3.7இல் காட்டப்பட்டுள்ளது.



படம் 3.7. இன்டெல் 8251இன் உருவ வரைபடம்

$C/\overline{D}$ (கட்டுப்பாடு / தரவு). இது 0 எனில், தரவானது தரவு ஊர்தியில் அனுப்பப்படும். உயர்நிலையெனில், கட்டுப்பாடு சைகைகள் தரவு ஊர்தியில் அனுப்பப்படும்.

$\overline{RD}$ (படி). இது 0 எனில், CPU இன்டெல் 8251லிருந்து தரவைப் படிக்கும்.

$\overline{WR}$ (எழுது). இது 0 எனில், இன்டெல் 8251இல் CPU தரவை எழுதும்.

$\overline{\text{DSR}}$ தரவுத்தொகுதி தயார்.

$\overline{\text{DTR}}$ தரவு முனையம் தயார்.

$\overline{\text{RTS}}$ அனுப்புவதற்கான வேண்டுகோள்.

$\overline{\text{CTS}}$ (காலி செய்வதற்கான விண்ணப்பம்). இந்த முனை 0 ஆனால், 8251ஐ செயலியக்கமடையச் செய்து தொடர் தரவை அனுப்பும்.

$\overline{\text{TxC}}$ (பரப்பி கடிகாரம்). இது தரவு அனுப்பும் விகிதத்தைக் கட்டுப்படுத்தும்.

T x E (வெற்றிடப் பரப்பி). 8251இல் எழுத்துக்கள் (பண்புகள்) அனுப்புவதற்கில்லையெனில், உயர்நிலைக்குச் செல்லும்.

TxRDY பரப்பி தயார்.
RxTDY ஏற்பி தயார்.
RxD தரவினை ஏற்றுக்கொள்வதற்கான வரி.
TxD தொடர் தரவினை அனுப்புவதற்கான வரி.
RxC (ஏற்பி கடிகாரம்). இது எந்த விகிதத்தில், எழுத்துக்கள் ஏற்றுக்கொள்ளப்படுகின்றன என்பதைக் கட்டுப்படுத்தும்.

உயர் **TxRDY** வரி, பரப்பி தரவினை ஏற்றுக்கொள்வதற்குத் தயார் என்று **CPU**க்குத் தெரியப்படுத்துகிறது. **CPU**லிருந்து தரவினை ஏற்றுக்கொண்டவுடன், $\overline{\text{CTS}}$ தாழ்நிலை எனில், 8251 தொடர் தரவினை **TxD** வரியில் அனுப்பும். மேலும், பரப்பியானது **CPU** வினால், செயலியக்கம் பெறுகிறது. **TxC** இடத்திலுள்ள கடிகார உள்ளீடு, தொடர் தரவு அனுப்புகின்ற விகிதத்தைக் கட்டுப்படுத்தும். ஒத்தியங்கு செலுத்தீட்டு முறையில் (**Synchronous Transmission Mode**), பட்டை விகிதம் கடிகார அதிர்வெண்ணிற்குச் சமமாகும். குரல் இசைவற்ற செலுத்தீட்டு முறையில் (**Asynchronous Transmission Mode**), பட்டை விகிதம் கடிகார அதிர்வெண்ணின் ஒரு பகுதியாகும். பகுதியானது 1, 1/16, அல்லது 1/64 ஆகும். தொடர் தரவானது **RxD** வரியின் வழியாகப் பெறப்படும். பொதுவாக **RxTDY** தாழ்நிலையிலிருக்கும். தொடர் தரவானது 8251இன் மூலம்

பெறப்பட்டவுடன், அது உயர்நிலையை அடையும். அது CPUக்கு 8251இல் செய்தி உள்ளது என்பதையும், CPUக்கு அனுப்புவதற்குத் தயார் நிலையில் உள்ளது என்பதையும் தெரியப்படுத்தும்.

$\overline{DSR}$, $\overline{DTR}$, $\overline{RTS}$ மற்றும் $\overline{CTS}$ இவைகள், திட்டமான மோடம் கட்டுப்பாட்டுச் சைகைகளாகும். இந்தக் கட்டுப்பாட்டு வரிகள், பயன்படுத்தப்படுகின்ற மோடத்தின் ஒத்தவரிகளுடன் இணைக்கப் படுகின்றன. மோடமானது ஒரு தகவேற்ற / தகவிறக்கக் கருவி ஆகும். இது ஏந்தி அதிர்வெண்ணைத் (Carrier Frequency) தரவு அனுப்புவதற்காகச் சேர்த்துக் கொள்ளும். உயர் அதிர்வெண் வழிமூலம் பெறப்பட்ட தரவினை, இந்த ஏந்தி அதிர்வெண் வெளியேற்றுகிறது.

மோடம் தரவினை அனுப்புவதற்கு தயார்நிலையில் உள்ளதைத் தெரியப்படுத்துவதற்காக, 8251 தாழ் $\overline{DTR}$ சைகையை அனுப்பும். $\overline{DSR}$ஐ தாழ்நிலையாக்கும்போது, தொடர்பியல் அமைப்பிற்குத் தரவினை முன்நோக்கி அனுப்புவதற்குத் தயாராக உள்ளது என்று மோடமும் பதிலளிக்கும். மோடமும் USART இரண்டும் தரவு அனுப்புவதற்குத் தயாராக இருக்கும்போது, 8251 தாழ் $\overline{RTS}$ சைகையைத் தரவு அனுப்புதலுக்கு முதலில் சேர்த்துக் கொள்வதற்காக மோடமிற்கு அனுப்புகிறது. 8251இன் இந்த மனுவை ஒப்புக்கொண்டதாக மோடமும் $\overline{CTS}$ஐ தாழ்நிலையாகக்குகிறது. தாழ் $\overline{CTS}$ 8251இன் அனுப்புதலின் வாதியலைச் செயலியக்குகிறது. தரவு அனுப்புதல் சில காரணங்களுக்காக நிறுத்தப்பட்டால், $\overline{CTS}$ உயர்நிலைக்குச் செல்லும்.

3.6 திட்டமிடத்தக்க எண்ணி / இடைவெளி நேரச்சுற்று :

பழக்கமான இடைவெளி நேரச்சுற்றிற்கான சில்கள், இன்டெல் 8253 மற்றும் 8254 ஆகும். 8253 d.c அதிர்வெண்ணின் விகிதத்திலிருந்து 2.6 MHz வரை செயல்படும். ஆனால், 8254 ஆனது, d.c அதிர்வெண்ணின் விகிதத்திலிருந்து 10 MHz வரை செயல்படும். 8253, NMOS தொழில்நுட்பவியலைப் பயன்படுத்தியது. 8254 HMOS தொழில் நுட்பவியலைப் பயன்படுத்தியது. இரண்டும் முனைக்கு முனை ஒத்திருக்கும். கீழ்க்கண்ட ஆறு முறைகளில் வேலை செய்யும்.

முறை 0 : முனையம் கணக்கீடு மீது குறுக்கீடு.

(Interupt on Terminal Count)

முறை 1 : திட்டமிடத்தக்க ஒரு எறிகை.(Progaramable one – shot)

முறை 2 : தகைவேக மின்னாக்கி (Rate Generator)

முறை 3 : சதுர அலையியற்றி (Square Wave)

முறை 4 : மென்பொருள் தொடக்கி முறை

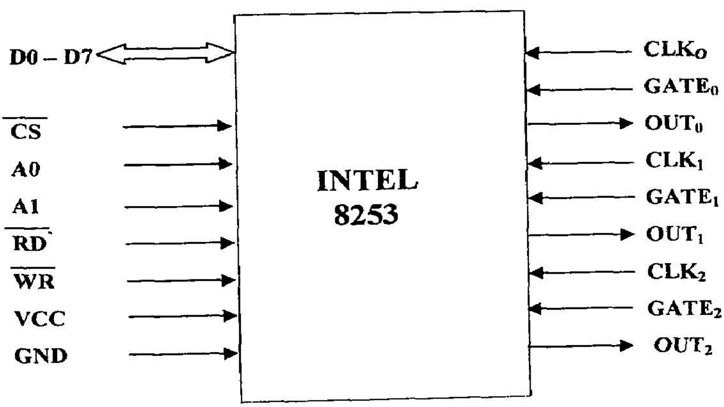
(Software Triggered Mode)

முறை 5 : வன்பொருள் தொடக்கி முறை

(Hardware Triggered Mode)

8086, 8088, 8085 மற்றும் பிற நுண்முறைப்படுத்திகளுக்கு, 8254 ஏற்றதாகும். 8253 ஆனது 8085 நுண்முறைப்படுத்திக்கு ஏற்றதாகும். 8254, 8253 ஐ விட உயர்ந்ததாகும்.

3. 6.1. இன்டெல் 8253



படம் 3.8. இன்டெல் 8253இன் உருவ வரைபடம்

8253 என்பது, 24 முனைகளைக் கொண்ட சில்லாகும். 5V$_{d.c.}$யில் வேலை செய்யும். இது மூன்று எந்திர 16 பிட் எண்ணிகளை கொண்டுள்ளது. திட்டநிரலாளி, 8253–லில் திட்டநிரல் செய்து

ஏதாவதொரு 6 வேலை செய்யும் முறைகளில் ஒன்றில் வேலை செய்ய வைக்கலாம். இது மென்பொருள் கட்டுப்பாட்டின் துணைகொண்டு வேலை செய்யும். முக்கியமான முனைகளின் விளக்கம், கீழே கொடுக்கப்பட்டுள்ளது.

$\overline{\text{RD}}$ (படி). இந்த முனை '0' (தாழ்) எனில், **CPU** தரவினைப் படிக்கும்.

$\overline{\text{WR}}$ (எழுது). இந்த முனை '0' (தாழ்நிலை) எனில், **CPU** தரவினை – செய்தி முறையில் அல்லது எண்ணிகளில் சுமையேற்றியும் வெளியிடும்.

A_0, A_1 இந்த முனைகள், முகவரி ஊர்தியுடன் இணைக்கப்பட்டுள்ளன. மூன்று எண்ணிகளில் ஏதாவது ஒன்றை தேர்வு செய்வதற்குப் பயன்படுகிறது. மேலும், மாதிரித் தேர்வு முறையில் கட்டுப்பாட்டுச் சொல் பதிவேட்டை எழுதுவதற்கும் பயன்படும்.

$\overline{\text{CS}}$ சில் தேர்வி.

D_0-D_7 இரு திசைகளின் தரவு ஊர்தி.

எண்ணி 0, எண்ணி 1 மற்றும் எண்ணி 2 என்பன முறையே CLK_0, CLK_1 மற்றும் CLK_2 ற்கான கடிகாரமாகும். $GATE_0$, $GATE_1$ மற்றும் $GATE_2$ என்பன முறையே எண்ணி 0, எண்ணி 1 மற்றும் எண்ணி 2க்கான கதவு முனைகள். எண்ணி 0, எண்ணி 1 மற்றும் எண்ணி 2க்கான வெளியீடு முனைகள் முறையே OUT_0, OUT_1 மற்றும் OUT_2 ஆகும்.

8253 தரவு ஊர்தித் தாங்கி, படி / எழுது வாதியல் மற்றும் கட்டுப்பாட்டுச் சொல் பதிவேடைக் கொண்டிருக்கும்.

தரவு ஊர்தித் தாங்கி :

இந்தத் தாங்கி 8253னுள்ளே இருக்கும். இது 3 நிலைகளைக்கொண்டது. இது ஒரு இரு திசையிலான 8 பிட் தாங்கி. இதனைப் பயன்படுத்தி 8253ஐ D_0 – D_7 வரிகள் வழியாகத் தரவு ஊர்தி அமைப்பு மூலம் இடைப்பரிமாறல் செய்யலாம்.

படி / எழுது வாதியல் :

8253இல் படி / எழுது வாதியல் உள்ளது. இது அமைப்பு ஊர்தியிலிருந்து உள்ளீட்டைப் பெற்று 8253–ஐ வேலை செய்வதற்காகக் கட்டுப்பாட்டுச் சைகைகளை உருவாக்குகிறது.

படி / எழுது வாதியலிற்கான பல்வேறு கட்டுப்பாடுகளின் தகுநிலை முனைகளின் இணைப்பு அட்டவணை 3.1இல் காட்டப்பட்டுள்ளது.

அட்டவணை : 3.1

$\overline{CS}$	A_1	A_0	$\overline{RD}$	$\overline{WR}$	
0	0	0	0	1	எண்ணி எண் 0வைப் படி
0	0	1	0	1	எண்ணி எண் 1ஐ படி
0	1	0	0	1	எண்ணி எண்2ஐ படி
0	0	0	1	0	எண்ணி எண் 0வை சுமையேற்று
0	0	1	1	0	எண்ணி எண் 1ஐ சுமையேற்று
0	1	0	1	0	எண்ணி எண் 2ஐ சுமையேற்று
0	1	1	1	0	எழுது முறை சொல்
0	1	1	0	1	வேலை இல்லை 3– நிலை
0	X	X	1	1	வேலை இல்லை 3 – நிலை
1	X	X	X	X	செயல் முடக்கம் 3 – நிலை

குறிப்பு : X என்பது வரையறுக்கப்படாத நிலையைக் குறிக்கும். நிலை 0 அல்லது 1 ஆக இருந்தாலும் முக்கியமில்லை என்பதே இதன் பொருள்.

கட்டுப்பாட்டுச் சொல் பதிவேடு :

A_0, A_1 என்ற முனைகள் முறையே 11 எனில், கட்டுப்பாட்டுச் சொல் தேர்வு செய்யப்படும். பதிவேட்டில் உள்ள கட்டுப்பாட்டுச் சொல்லின் வடிவ அமைப்பு, கீழே காட்டப்பட்டுள்ளது,

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	RL1	RL0	M2	M1	M0	BCD

கட்டுப்பாட்டுச் சொல்லிலுள்ள D_7 மற்றும் D_6 பிட்டுகள், மூன்றில் ஏதாவது ஒரு எண்ணியைத் தேர்வு செய்யும். D_5 மற்றும் D_4 என்பன முறையே எண்ணியைச் சுமையேற்றுவதற்கும் / படிப்பதற்கும் பயன்படுகின்றன. D_3, D_2 மற்றும் D_1 என்பன தேர்வு செய்யப்பட்ட எண்ணியில், செயற்பாட்டு முறையைத் தேர்வு செய்யப் பயன்படுகின்றன. 8253லுள்ள ஒவ்வொரு எண்ணியிலும் ஆறு செயற்பாட்டு முறைகள் உள்ளன. விரும்பப்பட்ட முறையில் செயல்படுவதற்காகத் தேர்வு செய்யப்பட்ட எ்ண்ணிக்குத் திட்டநிரல் வரையலாம். செயல்பாடுகளின் ஆறு முறைகள் யாவன : முறை 0, முறை 1, முறை 2, முறை 3, முறை 4 மற்றும் முறை 5. ஈரடி அல்லது BCD எண்ணிக்கைகளைத் தேர்வு செய்வதற்காக உள்ள பிட் D_0, D_0 '0' எனில், தேர்வு செய்யப்பட்ட எண்ணி ஈரடி எண்ணியில் செயல்படும். அதற்கு மாறாக 1 எனில், தேர்வு செய்யப்பட்ட எண்ணி, BCD எண்ணியில் செயல்படும்.

8253–இல் உள்ள எண்ணிகளின் செயல்பாடுகள், மென்பொருள் வல்லுநரால் திட்டநிரல் செய்யப்பட்டிருக்கும். கட்டுப்பாட்டுச் சொல் எண்ணியைத் தேர்வு செய்தல், அதன் செயற்பாட்டு முறை, சுமையேற்றம், எண்ணிக்கையின் வரிசை மற்றும் ஈரடி அல்லது BCD எண்ணிக்கையின் தேர்வு, இவற்றைத் (கட்டுப்பாட்டுச் சொல்) தீர்மானிக்கின்றது. கட்டுப்பாட்டுச் சொல் பதிவேட்டில் கட்டுப்பாட்டுச் சொல்லை எழுதியவுடனே, விருப்பப்பட்ட எண்ணியானது தேர்வு செய்யப்படுகிறது. செயற்பாட்டுமுறை அமைவு செய்யப்படுகிறது எண்ணிக்கையின் வரிசையின் சுமையேற்றம் மற்றும் ஈரடி அல்லது எண்ணிக்கையின் தேர்வு யாவும் வரையறுக்கப்படுகின்றன.

SC (எண்ணித் தேர்ந்தெடு) :

எண்ணியைத் தேர்ந்தெடுப்பதற்கு, **SC0** மற்றும் **SC1** முறையே கீழ்க்கண்டவாறு அமைக்கப்படுகின்றன.

SC1	SC0	
0	0	எண்ணி 0 தேர்ந்தெடு
0	1	எண்ணி 1 தேர்ந்தெடு
1	0	எண்ணி 2 தேர்ந்தெடு
1	1	தெளிவற்றது.

RL (படி / சுமை)

சுமை / படி எண்ணிகளைத் தேர்ந்தெடுப்பதற்கு, **RL0** மற்றும் **RL1** கீழ்க்கண்டவாறு அமைக்கப்படுகின்றன.

RL1	RL0	
0	0	எண்ணியின் செயற்பாட்டைத் தாழிடல்
0	1	படி / சுமை மிகச்சிறிய முக்கியமான பைட் மட்டும்
1	0	படி /சுமை மிக அதிக முக்கியமான பைட் மட்டும்
1	1	படி / சுமை மிகச்சிறிய முக்கியமான பைட்டை முதலில், பின்னர் மிக அதிக முக்கியமான பைட்.

முறை :

முறையினைத் தேர்ந்தெடுக்கும் பிட்டுகள் **M0, M1** மற்றும் **M2** கீழ்க்கண்டவாறு அமைக்கப்படுகின்றன.

M2	M1	M0	முறை
0	0	0	0
0	0	1	1
X	1	0	2
X	1	1	3
1	0	0	4
1	0	1	5

BCD

0 16 பிட்டுகள் இரும எண்ணி

1 இருமக் குறியீட்டுப் பதின்ம (**BCD**) எண்ணி

3. 6.2. கணக்கிடும் போது படித்தல் : (Reading While Counting)

எண்ணியில் உள்ள எண்ணிக்கையைப் படித்துக்கொண்டிருக்கும்போது, அதனைத் தாழிடுவதற்கு ஒரு கட்டளை உண்டு. செயற்பாட்டைத் தாழிடுவதற்கான கட்டுப்பாட்டுச் சொல்லின் பிட் மாதிரி, கீழே கொடுக்கப்பட்டுள்ளது.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	0	0	X	X	X	X

SC1 மற்றும் **SC0**	–	எண்ணித் தாழிடுவதைக் குறிப்பிடுவது
D_5 மற்றும் D_4	–	00 என்பது எண்ணியில் தாழிடல் செயற்பாட்டை நிர்மானிக்கும்.
X	–	தடை செய்யப்பட்டதைக் குறிக்கும்.

தாழிடல் கட்டளையைப் பெற்றவுடன், 8253 தேர்ந்தெடுக்கப்பட்ட எண்ணியில் உள்ளவற்றைச் சேமிப்புப் பதிவேட்டில் தாழிடும். தற்பொழுது, நுண்கணிப்பொறி சேமிப்புப் பதிவேட்டில் உள்ளவற்றை ஒரு சாதாரண படி கட்டளையை வெளியிட்டுப் படிக்கும். மூல முறை கட்டளை அமைவில் **RL1** மற்றும் **RL0** பிட்டுகளைக் கொண்டு படித்தல் செயல் நடைபெறும். அதாவது **LSB**-இன் எண்ணிக்கை மட்டும் அல்லது **MSB**-இன் எண்ணிக்கை அல்லது முதலில் **LSB**-இன் எண்ணிக்கை, பின் **MSB**-இன் எண்ணிக்கையைப் படிப்பதற்கான செயல் நடைபெறும். நுண்கணிப்பொறி முதலில் எண்ணியில் உள்ளவற்றைத் தாழிட்டுப் பின்னர் அதனைப் படிப்பதற்குக் கோரும்போது தாழிடல் கட்டளையை ஒவ்வொரு நேரத்திலும் வெளியிடும். எண்ணிகளின் முறை அல்லது அதன் செயற்பாடுகள், தாழிடல் கட்டளையினால் பாதிக்கப்படுவதில்லை.

3.6.3. <u>செயற்பாடு</u> :

8253–இல் மூன்று தனித்த இறங்கு விளிம்புத் துவக்கியால் இயங்கக்கூடிய 16 பிட் இறங்கு எண்ணிகள் உள்ளன. அவையாவன, எண்ணி 1 மற்றும் எண்ணி 2. எண்ணிகள் யாவும் முழுமையாகத் தனித்துச் செயல்படக் கூடியன என்பதால், ஒவ்வொரு எண்ணியும் வெவ்வேறு வகையான உருவமைப்புடைய எண்ணிக்கைச் செயற்பாட்டையும் பெற்றிருக்கும். பொதுவாக, நுண்முறைப்படுத்தியில் 8253 இருக்கும். இதில் ஒரு எண்ணியானது படி செயற்பாட்டிற்குப் பயன்படும். எண்ணி 1 மற்றும் எண்ணி 2 என்ற இரண்டு எண்ணிகளிலுள்ள முனைகள், இணைப்பு இடம் **P1** மற்றும் **P2** இல் பெறக்கூடியன. எண்ணி 0 ஒற்றைப்படி (**Single – Step Operation**) செயற்பாட்டிற்குப் பயன்படும். 8253–க்கான கடிகார அதிர்வெண், **2.6 MHz**ஐ விடக் குறைவாக இருக்கும். 8085இன் வெளியீட்டு கடிகார அதிர்வெண் சுமார் **3MHz**. ஒரு விளிம்புத் துவக்கி நிலைமாறி 7474 பயன்படுத்தப்பட்டு, கடிகார அதிர்வெண்ணை இரண்டாக பிரிக்கப் பயன்படுகிறது. அதனால், கடிகார அதிர்வெண் சுமார் **1.5 MHz** முறைப்படுத்தியில் கிடைக்கும். ஒவ்வொரு எண்ணியிலும் இணைப்பிடத்தில் கிடைக்கக் கூடிய மூன்று முனைகள் **CLK, GATE** மற்றும் **OUT** என்பன. வினிடிக்ஸ் முறைப்படுத்தியில் ஒரு எண்ணியில் இருக்கக்கூடிய **CLK** முனை **1.5MHz** கடிகாரத்துடன் இணைக்கப்பட்டுள்ளது. மற்ற எண்ணியில்

உள்ள CLK முனைகள் இணைக்கப்படவில்லை. பயனாளர் தன்னுடைய விருப்பத்திற்கு ஏற்றாற்போல், கடிகார அதிர்வெண்ணை இந்த எண்ணிற்காகப் பயன்படுத்தலாம். பயனாளர் எண்ணியினுடைய CLK முனையைப் பயன்படுத்துவதற்கு முன்னால் கண்டிப்பாகத் தணிக்கை செய்ய வேண்டும். அதாவது, அது முன்பாகவே கடிகாரத்துடன் இணைக்கப்பட்டுள்ளதா அல்லது தேவையான கடிகார அதிர்வெண்ணை இதில் கொடுக்க வேண்டுமா எனச் சரிபார்க்க வேண்டும்.

கட்டுப்பாட்டுச் சொல் பதிவேட்டிற்கான முகவரிகள் மற்றும் வினிடிக்ஸ் முறைப்படுத்தியிலுள்ள **8253**இன் எண்ணிகள், கீழே கொடுக்கப்பட்டுள்ளன.

எண்ணி	**0**	-	**10**
எண்ணி	**1**	-	**11**
எண்ணி	**2**	-	**12**

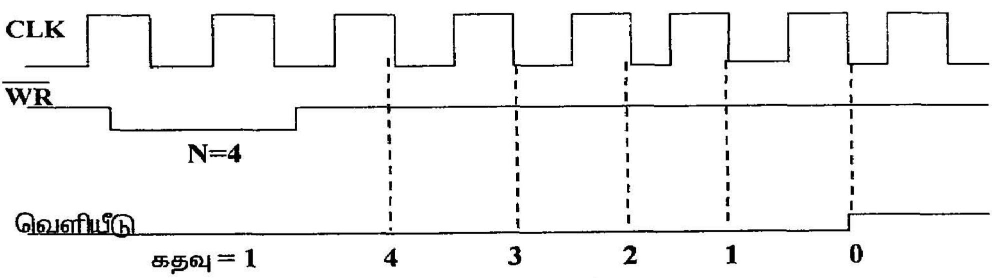
கட்டுப்பாட்டுச் சொல் பதிவேடு – **13**

எண்ணியானது, பல்வேறு பயன்பாடுகளுக்குப் பயன்படுத்தப் படுகின்றது. அதாவது, **BCD** / ஈரடி எண்ணி, திட்ட நிரலாக்கி, சதுரலையியற்றி, வன்பொருள் / மென்பொருள் தொடக்கி, நேரத்தாழ்வு உண்டாக்குவதற்குப் பயன்படுகின்றன.

3. 6.4. முறை 0 :

தடங்கல் முறை 0, துல்லியமான நேரத்தாழ்வை, மென்பொருள் கட்டுப்பாட்டிற்குக் கீழே உண்டாக்குவதற்குப் பயன்படுகிறது. **8253**–இல் உள்ள எண்ணிகளில் ஒன்றைத் தொடக்கி, விருப்பமான நேரத்தாழ்விற்கான சரியான எண்ணைச் சுமையேற்றலாம். விருப்பமான நேரத்தாழ்விற்குப் பிறகு நிறைவேற்ற வேண்டியதை, நுண்முறைப்படுத்தி தடங்கலின்போது நிறைவேற்றும்.

கட்டுப்பாட்டுச் சொல்லைக் கட்டுப்பாட்டுச் சொல் பதிவேட்டில் சுமையேற்றி, முறை அமைக்கப்படுகிறது. முறை 0 செயற்பாட்டில், முறை அமைக்கப்பட்ட பிறகு, எண்ணியின் வெளியீடு முதலில் தாழ்நிலையிலிருக்கும். அதன் பிறகு, விருப்பமான எண் N ஐக் கொண்டு எண்ணியைச் சுமையேற்றலாம். சுமையேற்றியவுடன், கதவு உயர்நிலையெனில், எண்ணிக்கை தொடங்கும். முறை 0 செயற்பாட்டில், கதவை உயர்நிலையில் வைக்கவேண்டும். எண்ணியில், எண்ணிக்கை சென்று கொண்டிருக்கும்போது வெளியீடு தாழ்நிலையிலிருக்கும். இறுதி எண்ணைத் தொட்டபின்பு, வெளியீடு உயர்நிலைக்குச் சென்று, எண்ணை மறுபடியும் சுமையேற்றும் வரையிலும் அல்லது புதிய எண்ணைச் சுமையேற்றும் வரையிலும் உயர்நிலையிலேயே இருக்கும். முறை 0 செயற்பாட்டின் நேர வரைபடம் படம் 3.9–இல் காட்டப்பட்டுள்ளது. எண்ணை மறு சுமையேற்றிய பின் அல்லது புதிய எண்ணைச் சுமையேற்றியவுடன், வெளியீடு தாழ்நிலைக்குச் செல்லும் மற்றும் எண்ணியானது எண்ணிக்கைச் செயற்பாட்டை மீண்டும் தொடக்கும்.



படம் 3.9 முறை 0, இறுதி எண்ணி மீது குறுக்கீடு

எண்ணிக்கைச் செயற்பாட்டின்போது, கதவைத் தாழ்நிலையாக நிர்மானித்தால் எண்ணிக்கை நின்றுவிடும். கதவு உயர்நிலைக்குத் திரும்பும்போது, எண்ணிக்கை மீண்டும் விட்ட இடத்திலிருந்து தொடங்கும். எண் மதிப்பை எப்பொழுது வேண்டுமானாலும் மாற்றலாம். எண்ணிக்கை தொடர்ந்து கொண்டிருக்கும்பொழுதே புதிய எண்ணைச் சுமையேற்றம் செய்யலாம். அதன் பிறகு, புதிய எண் மதிப்பிலிருந்து மீண்டும் எண்ணிக்கை தொடங்கும். புதிய எண் ஒரு பைட் நீளம் இருந்தால், எண்ணிக்கை உடனடியாக புதிய எண்ணிலிருந்து தொடங்கும். புதிய எண் இரண்டு பைட் நீளம்

கொண்டதாகயிருந்தால், புதிய எண்ணின் முதல் பைட் முடிந்த உடன், எண்ணியானது எண்ணிக்கையை நிறுத்தி விடும். இரண்டாவது பைட் சுமையேற்றியவுடன், எண்ணி மீண்டும் எண்ணிக்கையைத் தொடங்கும்.

வெளி முனையானது, நுண்முறைப்படுத்தியைத் தடங்கல் செய்வதற்குப் பயன்படும். இந்த வகைச் செயற்பாடுகள் சரியான நேரத்தாழிடலை உண்டாக்குவதற்குப் பயன்படுகின்றன.

<u>உதாரணம்</u> : 1

8253 ஐப் பயன்படுத்தி, ஒரு குறிப்பிட்ட திட்டமான வேலையைச் சிறிது தாழ்த்தலிற்குப் பிறகு செய்வதற்கு, **8253** இன் முறை **0** செயற்பாடு பயன்படுகிறது. தாழ்த்தலிற்கான எண் **FF50** என எடுத்துக்கொண்டால், எண்ணிக்கை முடிந்தவுடன், **FD01** மற்றும் **FD02** என்ற நினைவக இருப்பிடங்களிலுள்ளவற்றை **FC31** மற்றும் **FC32** என்ற நினைவக இருப்பிடங்களுக்கு மாற்றுவதற்கும், நுண்முறைப்படுத்தியைக் குறுக்கீடு செய்யவதற்கும் எண்ணி தேவைப்படுகிறது.

அனைத்துக் குறுக்கீடுகளும் செயலியக்கம் பெறுகின்றன. **RST 7.5, 6.5** மற்றும் **5.5** ஆகியவற்றைச் செயலியக்க, **SIM**-ற்கான சேமக்கலத்தில் உள்ளவை, கீழ்க்கண்டவாறு திட்டநிரல் செய்யப்பட்டுள்ளன.

7	6	5	4	3	2	1	0
S0D	**S0E**	**X**	**R7.5**	**MSE**	**M7.5**	**M6.5**	**M5.5**
0	**0**	**0**	**0**	**1**	**0**	**0**	**0=08**

முறை 0 செயற்பாட்டிற்கான கட்டுப்பாட்டுச் சொல்

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	1	1	1	0	0	0	0=70 H

எண்ணி 1ஐ தொடக்க நிலைக்குக் கொண்டு வருவதற்காக D_7 மற்றும் D_6 ஐ, 01 என்ற நிலையில் அமைக்க வேண்டும். D_5 மற்றும் D_4 ஐ முதலில் 11 எனக் கொண்டு, எண்ணின் மிகச்சிறிய முக்கியமான பிட்டைச் (LSB) சுமையேற்றிய பிறகு, மிக முக்கியமான பிட்டைச் (MSB)சுமையேற்றும். செயல்வகை 0 விற்காக D_3, D_2, D_1 எல்லாம் 000 என அமைக்க வேண்டும். D_0வை 0 என அமைத்து, எண்ணிக்கையை ஈரடியில் மாற்றவேண்டும். எண்ணி 1–இன் வெளி முனையை RST 7.5 உடன் இணைக்க வேண்டும்.

<u>திட்டநிரல்</u> :

நினைவக முகவரி	இயந்திரக் குறியீடு	குறிப்பு செயற்	இயமிகள்	விளக்கம்
FC00	FB	EI		அனைத்துக் குறுக்கீடுகளும் செயலியக்கம் பெறும்.
FC01	3E,08	MVI	A,08	RST 7.5, 6.5 மற்றும் 5.5 செயலியக்கம் பெற, பிட் அமைப்பைச் சுமையேற்று.
FC03	30	SIM		RST 7.5, 6.5 மற்றும் 5.5 ஐ செயலியக்கமடையச் செய்.
FC04	3E,70	MVI	A,70	எண்ணி 1 ஐ ஆரம்ப நிலைக்குக் கொண்டு செல்ல, முறை 0க்கான கட்டுப்பாட்டுச் சொல்.
FC06	D3,13	OUT	13	கட்டுப்பாட்டுச் சொல் பதிவேட்டில் எழுது.

FC08	**3E,50**	**MVI**	**A,50 H**	எண்ணிக்கையின் மிகச்சிறிய முக்கியமான பைட்.
FC0A	**D3,11**	**OUT**	**11**	**LSB** ஐக் கொண்டு எண்ணி 1 ஐ சுமையேற்று.
FC0C	**3E, FF**	**MVI**	**A,FF**	எண்ணிக்கையின் **MSB**
FC0E	**D3, 11**	**OUT**	**11**	எண்ணின் **MSB** கொண்டு எண்ணியைச் சுமையேற்று.
FC10	**C3, 10, FC**	**JMP**	**HERE**	

RST 7.5 நுண்முறைப்படுத்தியைக் குறுக்கீடு செய்யும்போது, திட்டநிரலானது நினைவக இருப்பிடம் **003C**-க்குத் தாவும். அதன்பின் மானிட்டர், திட்டநிரலை **003C** இருப்பிடத்திலிருந்து **FFBD** இருப்பிடத்திற்கு மாற்றும். தற்போது, பயனாளர் திட்டநிரலில் **FFBD**லிருந்து சேவை துணை நிரற்செயல்முறையின் தொடக்க முகவரிக்கு மாற்ற வேண்டும். இதில் சேவை துணை நிரற்செயன்முறை, **FC20** நினைவக இருப்பிடத்திலிருந்து ஆரம்பமாகிறது.

FFBD	**C3, 20, FC**	**JMP**	சேவை துணை நிரற்செயன்முறைக்குத் தாவு.

சேவை துணை நிரற்செயன்முறை :

FC20	**2A, 01, FD**	**LHLD**	**FD01**	**FD01** மற்றும் **FD02** உள்ளவற்றை **H-L** இணையில் பெறு.
FC23	**22,31,FC**	**SHLD**	**FC31**	அவற்றை **FC31** மற்றும் **FC32** இல் சேமி.
FC26	**FB**	**EI**		அனைத்துக் குறுக்கீடுகளையும் செயலியக்கம் பெறச் செய்.
FC27	**C9**	**RET**		

தரவு

FD01 – 5B
FD02 - 38

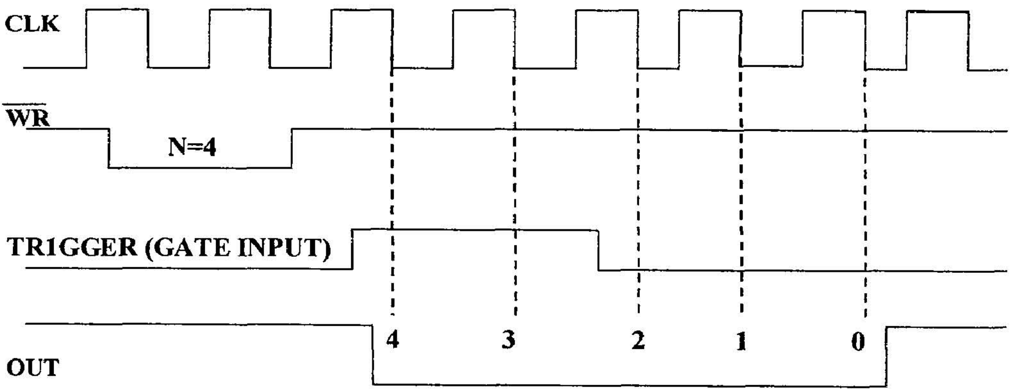
முடிவு

FC31 – 5B
FC32 - 38

<u>3. 6.5. முறை1</u> : திட்டமிடத்தக்க ஒரு தாக்கம். (Programmable one shot)

முறை 1இல் எண்ணி மறுதுவக்கியாகவும், ஒரு மோதல் திட்டநிரலியாகவும் வேலை செய்யும். தாழ்நிலையிலிருந்து உயர்நிலைக்கு மாறுகின்ற சைகையைக் கதவில் கொடுக்கும்போது, அது துவக்கிச் சைகையாக வேலை செய்யும். இந்த வகையான செயற்பாட்டில், முறை அமைவிற்குப் பின் வெளியீடு முதலில் உயர்நிலையிலிருக்கும். செயற்பாட்டிற்காக முறை அமைவிற்குப் பின்னர், எண்ணி எண் N என்ற மதிப்பால் சுமையேற்றப்படும். பின்னர் எண்ணி, எண்ணிக்கையில் குறையத் தொடங்கும். N கடிகாரச் சுழற்சிகளின் ஒவ்வொரு தாழ்நிலையிலிருந்தும் உயர்நிலைக்கு மாறும் சைகை கதவு உள்ளீட்டில் வரும்போது, வெளியீடானது தாழ்நிலைக்குச் செல்லும். கதவு உள்ளீட்டில்

சைகையின் ஏறு விளிம்பிற்குப் பின், முதல் எதிர்விளிம்பு கடிகார இடத்தில் வெளியீடானது தாழ்நிலைக்குச் செல்லும். இதனைப் படம் 3.10இல் காணலாம்.



படம் 3.10 முறை 1

எண்ணிக்கை தொடர்ந்து கொண்டிருக்கும்பொழுது, வெளியீடு தாழ்நிலையிலேயிருக்கும். கடைசி எண்ணிக்கையின்போது, அது உயர்நிலைக்குச் செல்லும். (எண்ணியில் உள்ளவை = 0). N ஐ மாற்றி, வெளியீடு துடிப்பின் அகலத்தை மாற்றலாம். வெளியீடு துடிப்பின் அகலத்திற்குத் திட்டநிரல் செய்யலாம். அதனால், இந்த முறைச் செயற்பாட்டிற்குத் திட்டமிடத்தக்க ஒரு தாக்கம் என்று பெயர்.

கதவு உள்ளீட்டை மீண்டும் தாழ்நிலையிலிருந்து உயர்நிலையாக்கும்போது, எண்ணி எண் மதிப்பீடால் மறு சுமையேற்ற வேண்டும். எண்ணி, மீண்டும் எண்ணிக்கையில் குறைந்துகொண்டே வரும். மற்றும் N கடிகாரத் துடிப்புகளுக்கு, வெளியீடு தாழ்நிலைக்குச் செல்கிறது. அதனால் ஒரு தாக்கு வகையான செயற்பாடு, மறு துவக்கியாகும்.

வெளியீடு தாழ்நிலையிலிருக்கும்போதே புதிய எண் சுமையேற்றப்பட்டால், அடுத்து வருகின்ற துவக்கி வரைக்கும், ஒரு தாக்குத் துடிப்பின் இடைவெளியில் மாற்றமிருக்காது. தாழ்நிலையிலிருந்து உயர்நிலைக்கு மாறும்போது, கதவு உள்ளீடு ஒரு எண்ணியைத் தொடக்க நிலையில் பெற்றிருந்தால், வெளியீடு குறையும் வரையிலும் எண்ணிக்கை

தொடர்ந்துகொண்டேயிருக்கும். இப்பொழுது மற்றொரு இதே போன்ற சைகை கதவில் தோன்றினால், இது வெளியீட்டைத் தாழ்நிலையிலேயே N கடிகாரத்துடிப்புகளுக்கு இருக்கச் செய்யும்.

உதாரணம் :

8053இல் செயல்வகை 1 ஐ பயன்படுத்தி, CROஇல் வெளியீட்டின் அலை உருவத்தைக் காண்க.

எண்ணி 1–க்கான கட்டுப்பாட்டுச் சொல் மற்றும் செயல்வகை 1 ஐ, கீழ்க்கண்டவாறு கண்டுபிடிக்கலாம்.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	1	0	1	0	0	1	0=52 H

எண்ணி 1–ஐ தேர்வு செய்ய, D_7 மற்றும் D_6 ஐ 01 ஆக அமைக்கவும்.

எண்ணின் LSB சுமையேற்றம் செய்ய, D_5 மற்றும் D_4 ஐ 01 ஆக அமைக்கவும்.

முறை 1ற்காக D_3, D_2, மற்றும் D_1ஐ 001 என வைக்கவும்.

ஈரடி எண்ணிக்கைக்காக D_0 ஐ 0 ஆக்கவேண்டும்.

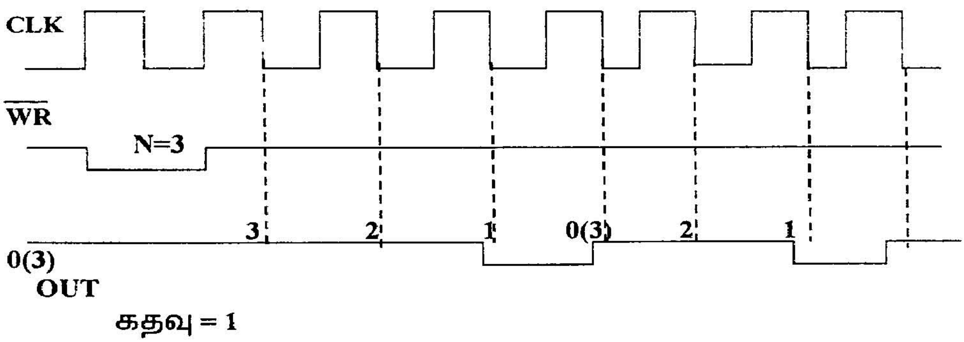
வெளியீடு அலையின் உருவம் மற்றும் கடிகாரம் (CLK), இவற்றை CROஇல் காண வேண்டும். PC0இன் முனையை 8253இன் கதவுடன் இணைக்க வேண்டும். நுண்முறைப்படுத்தியின் மூலம் துடிப்பை உருவாக்கி, அதனை PC0இல் வெளியிட வேண்டும். துடிப்பை 8253–ன் கதவிற்குக் கொடுக்க வேண்டும். முறை 1இல், கதவின் உள்ளீட்டில் தோன்றுகின்ற தாழ்நிலையிலிருந்து உயர்நிலைக்கு ஏற்படும் மாற்றத்தின்போது, எண்ணிக்கையானது தொடங்கும். ஆகவே, தூண்டுதல் சைகையைக் கதவிற்குத் தர வேண்டும். கணக்கீட்டைத் தாழ்நிலையில் நிறுத்த வேண்டும். ஏனெனில், எண்ணிக்கை முடிவதற்கு முன்னால், மற்றொரு துடிப்பு கதவிற்குக் கொடுக்கப்படும். அப்படியானால், போதுமான உயர்நிலை நிறுத்தத்தின்போது, திட்டநிரலானது FC14லிருந்து FC0Cக்குத் தாவுவதற்கு முன்னால், நேர தாமதம் சேர்த்துக் கொள்ளப்படும். துடிப்பின் அகலத்தைக் கணக்கீட்டின் மதிப்பிற்கு ஏற்றாற்போல் மாற்றலாம். திட்டநிரலில் வளையச்சுற்றைப் புகுத்தும்போது, சைகை துவக்கி மீண்டும் கதவிற்குக் கொடுக்கப்படுகிறது. இது சைகையைத் திரும்பச் செலுத்துவதில் முடிவதால், CROஇல் பார்க்க முடியும்.

திட்டநிரல் :

நினைவக முகவரி	இயந்திரக் குறியீடு	குறிப்பீடுகள்	செயற் குறிப்பு	இயமிகள்	விளக்கம்
FC00	3E, 52		MVI	A,52 H	செயல்வகை 1இல் எண்ணியைத் துவக்குவதற்காகக் கட்டுப்பாட்டுச் சொல்லை வாங்கு.
FC02	D3, 13		OUT	13	கட்டுப்பாட்டுச் சொல் பதிவேட்டில் எழுது.
FC04	3E, 08		MVI	A,08	எண்ணை வாங்கு
FC06	D3, 11		OUT	11	எண்ணி 1 ஐ எண்ணால் சுமையேற்று.
FC08	3E, 80		MVI	A,80 H	PPI 8255.2–ஐ ஆரம்ப நிலையாக்கு.
FC0A	D3, 0B		OUT	0B	
FC0C	3E, 00	LOOP	MVI	A,00	PC0இல் வெளியிடக் கூடிய துடிப்பை உண்டாக்கு. அதனை 8253இன்கதவுடன் இணைக்க வேண்டும்.
FC0E	D3, OA		OUT	0A	
FC10	3E, 01		MVI	A, 01	
FB12	D3, 0A		OUT	0A	
FC14	C3, 0C, FC		JMP	LOOP	

3. 6.6 முறை : 2 தகைவேக மின்னாக்கி

முறை 2இல் எண்ணி, ஒரு எளிய N ஆல் வகுபடு எண்ணியாக வேலை செய்யும். இந்த வகையை அமைத்தால், எண்ணியின் வெளியீடு முதலில் உயர்நிலையிலிருக்கும். அதன் பிறகு எண்ணி கணக்கீடு N ஆல் சுமையேற்றப்படும். முறை 2 செயற்பாட்டில், கதவை உயர்நிலையில் நிறுத்த வேண்டும். இந்த வகையில் வெளியீடானது (N-1), கடிகாரத் துடிப்புகளுக்கு உயர்நிலையிலேயிருக்கும். அதன்பின், ஒரு கடிகாரத் துடிப்பு தாழ்நிலைக்குச் செல்லுவது, படம் 3.11–இல் காட்டப்பட்டுள்ளது.



படம் 3.11. முறை 2, தகைவேக மின்னாக்கி

அதன் பின்னர் வெளியீடு மீண்டும் உயர்நிலைக்குச் சென்று, கணக்கீடு N தன்னிச்சையாக எண்ணியில் சுமையேற்றப்படும். மீண்டும் (N-1) துடிப்புகளுக்கு, வெளியீடு உயர்நிலையிலேயே இருக்கும். பின்னர் ஒரு கடிகாரத் துடிப்பிற்குத் தாழ்நிலைக்குச் செல்லும். கதவு உள்ளீடு உயர்நிலையிலிருக்கும் வரையிலும், இந்த மாதிரி முழுமையான செயல்முறை திரும்பத் திரும்ப நடைபெறும். கதவு உள்ளீட்டைத் தாழ்நிலையில் வைத்தால், எண்ணி செயல்முடக்கம் பெற்று, வெளியீடு உயர்நிலையில் இருக்கும். கதவு உயர்நிலைக்குத் திரும்பினால், எண்ணி இயல்பான செயற்பாட்டைத் திரும்ப எடுத்துக்கொண்டு, மீண்டும் முதல் கணக்கீட்டிலிருந்து தொடங்கும்.

புதிய கணக்கீட்டை, வெளியீடு துடிப்புகளுக்கிடையே எண்ணிப் பதிவேட்டில் சுமையேற்றத் தற்போதைய காலஅளவில் மாற்றம் இல்லாமல், பின்னர் வரும் கால அளவில் புதிய கணக்கீட்டின் மதிப்பு எதிரொலிக்கப்படும்.

உதாரணம் : 1

முறை 2இல் 8253ஐப் பயன்படுத்தி, அதனை N ஈரடி எண்ணியாக வேலை செய்ய வைக்கலாம். N=16 தசம எண் = 10 பதினாறடி எண். இதற்காக எண்ணி 2 பயன்படுத்தப்படுகிறது. கட்டுப்பாட்டுச் சொல்லைக் கீழ்க்கண்டவாறு கண்டுபிடிக்கலாம்.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	1	0	0=94 H

D_7 மற்றும் D_6 என்பன எண்ணியைத் தேர்வு செய்வதற்கான பிட்டுகள். அவைகளை 10இல் அமைத்து எண்ணி 2 ஐ தேர்வு செய்யலாம். D_5 மற்றும் D_4 ஐ பயன்படுத்திக் கணக்கீடு N ஐ சுமையேற்றலாம். அவற்றை 01இல் அமைத்துக் கணக்கீட்டின் மிகச்சிறிய முக்கியமான பிட்டை மட்டும் சுமையேற்றலாம். D_3, D_2 மற்றும் D_1 இவைகள் முறையைத் தேர்ந்தெடுப்பதற்கான பிட்டுகள். முறை 2 வேண்டுமெனில், அவற்றை 010 = 2 என அமைக்கவும்.

BCD அல்லது ஈரடி எண்ணிக்கைக்குரிய பிட் D_0 இது 0 வாக இருந்தால், ஈரடி எண்ணிக்கை ஆகும்.

முறை 2இன் செயற்பாட்டிற்கான கட்டுப்பாட்டுச் சொல் 94. இது எண்ணி 2க்காக ஈரடி எண்ணிக்கைக்காக நிச்சயிக்கப்பட்டது. 13 கட்டுப்பாட்டுச் சொல் பதிவேட்டிற்கான முகவரி. எண்ணி 2க்கான முகவரி 12. ஒவ்வொரு 16 கடிகாரத் துடிப்புகளுக்கொரு கடிகாரத்துடிப்பில், வெளியீடு தாழ்நிலைக்குச் செல்வதை CROஇன் மூலம் காணலாம். ஈரடி எண்ணிக்கைக்காகக் கணக்கீட்டு மதிப்பு N ஐ பதினாறடி எண்ணில் வைக்கவேண்டும்.

<u>திட்டநிரல்</u> :

நினைவக முகவரி	இயந்திரக் குறியீடு	செயற் குறிப்பு	இயமிகள்	விளக்கம்
FC00	3E, 94	MVI	A,94 H	எண்ணி 2, முறை 2, ஈரடி எண்ணிக்கைக்காக சொல்லை வாங்கு.
FC02	D3, 13	OUT	13	13 என்பது, நுண்முறைப் படுத்தியினுள் கட்டுப் பாட்டுச் சொல்லிற்காக எழுதக்கூடிய முகவரி.
FC04	3E, 10	MVI	A,10 H	கணக்கீடு மதிப்பு N ஐ வாங்கு 10 Hex = 16 தசம எண்.
FC06	D3,12	OUT	12	எண்ணி 2 க்கான முகவரி 12.
FC08	76	HLT		நிறுத்து.

<u>உதாரணம்</u> : 2

8253ஐ முறை 2இல் பயன்படுத்தி, ஒரு தகைவேக மின்னாக்கியாக வேலை செய்ய வைக்கலாம். எண்ணி ஒரு BCD எண்ணியாக வேலை செய்யும்.

N = 10 தசம எண்.

எண்ணி 2ஐ பயன்படுத்தி, முறை 2க்கான கட்டுப்பாட்டுச் சொல்.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1 .	0	1	0	1=95 H

எண்ணி 2 ஐ தேர்வு செய்வதற்காக, D_7 மற்றும் D_6 முறையே 10 என அமைக்கவும்.

கணக்கீட்டின் மிகச்சிறிய முக்கியமான பைட்டைச் சுமையேற்றுவதற்காக, D_5 மற்றும் D_4 ஐ 01 என வைக்கவும்.

முறை 2ற்காக D_3, D_2 மற்றும் D_1ஐ 010 = 2 என அமைக்கவும்.

BCD எண்ணிக்கையிக்காக D_0 ஐ 1 ஆக வைக்கவும்.

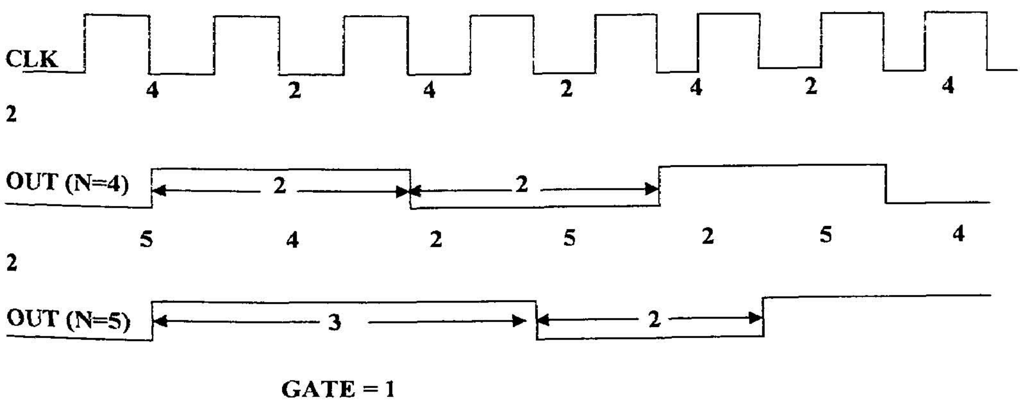
<u>திட்டநிரல்</u> :

நினைவக முகவரி	இயந்திரக் குறியீடு	செயற் குறிப்புகள்	இயமிகள்	விளக்கம்
FC02	**3E, 95**	**MVI**	**A,95 H**	எண்ணி 2 ஐ ஆரம்ப நிலைக்குக் கொண்டு வருவதற்காகக் கட்டுப்பாட்டுச் சொல்லை வாங்கு.
FC02	**D3, 13**	**OUT**	**13**	கட்டுப்பாட்டுச் சொல் பதிவேட்டில் எழுது.
FC04	**3E, 10**	**MVI**	**A,10**	கணக்கீடைத் தசம எண்ணில் வாங்கு **N = 10.**
FC06	**D3, 12**	**OUT**	**12**	எண்ணி 2 ஐ கணக்கீடுகள் கொண்டு சுமையேற்று.
FC08	**76**	**HLT**		நிறுத்து

BCD எண்ணிற்கான கட்டுப்பாட்டுச் சொல் 95. கணக்கீடு N=10ஐ தசம எண்ணில் எடுக்க வேண்டும். BCD எண்ணியில், வெளியீடு ஒவ்வொரு பத்துக் கடிகாரத் துடிப்புகளுக்கு ஒரு துடிப்பின்போது, தாழ்நிலைக்குச் செல்லும். BCD எண்ணிற்கான கட்டுப்பாட்டுச் சொல் என்பதால், 8253, Nஐ ஒரு BCD எண்ணாகக் கருதும். CLK மற்றும் வெளியீடு அலைகளை, CRO யில் காணலாம்.

3. 6.7. முறை : 3 சதுரலை இயற்றிகள் :

முறை 3இல் எண்ணி, சதுரலை இயற்றியாக வேலை செய்யும். இந்த வகை செயற்பாட்டில் அமைத்தவுடன், எண்ணி கணக்கீட்டு மதிப்பு N ஆல் சுமையேற்றப்படுகிறது. முறை 3இன் செயற்பாட்டிற்காகக் கதவை உயர்நிலையில் வைக்க வேண்டும். Nஇன் மதிப்பு இரட்டைப் படையாக இருந்தால், N/2 கடிகாரத் துடிப்புகளுக்கு வெளியீடு, உயர்நிலையிலேயே இருக்கும். அடுத்த N/2 கடிகாரத் துடிப்புகளுக்கு வெளியீடு தாழ்நிலைக்குச் செல்லும். Nஇன் மதிப்பு ஒற்றைப்படையாக இருந்தால், (N+1) /2 கடிகாரத் துடிப்புகளுக்கு வெளியீடு உயர்நிலையிலும், அடுத்த (N+1) /2 கடிகாரத் துடிப்புகளுக்குத் தாழ்நிலைக்கும் செல்லும். கதவைத் தாழ்நிலையில் வைத்தால், எண்ணி செயல் முடக்கம் அடைந்து வெளியீடு உயர்நிலையில் இருக்கும். கதவு உயர்நிலைக்கு வரும்பொழுது, எண்ணி இயல்பான செயற்பாட்டிற்குத் திரும்பும். அலை உருவப்படம், 3.12–இல் காட்டப்பட்டுள்ளது.



படம் 3.12 முறை 3, சதுரலை இயற்றி

உதாரணம் : 1

8253இல் முறை 3 யைப் பயன்படுத்திச் சதுரலை இயற்றியாக மாற்றலாம்.

N=16 எண்ணியானது, ஈரடி எண்ணியாகச் செயற்படும். எண்ணி 2, இதற்காகப் பயன்படுகிறது. கட்டுப்பாட்டுச் சொல்லைக் கீழ்க்கண்டவாறு முடிவு செய்யலாம்.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	1	1	0=96 H

எண்ணி 2 ஐ தேர்வு செய்ய, D_7 மற்றும் D_6 முறையே 10 என அமைய வேண்டும்.

கணக்கீட்டின் LSBஐ சுமையேற்ற, D_5 மற்றும் D_4 01 ஆக இருக்க வேண்டும்.

முறை 3ற்காக, D_3, D_2 மற்றும் D_1ஐ 011 என அமைக்க வேண்டும்.

ஈரடி எண்ணிக்கைக்காக D_0 வை 0 வாக மாற்ற வேண்டும்.

N=16 எனில், 8 கடிகாரச் சுற்றுகளுக்கு வெளியீடு உயர்நிலையிலும், பின்னர் அடுத்த 8 கடிகாரச் சுற்றுகளுக்குத் தாழ்நிலையிலும் இருக்கும். N=17 (ஒற்றைப்படை எண்) எனில், முதல் 8 கடிகாரச் சுற்றுகளுக்கு வெளியீடு உயர்நிலையிலும், பின்னர் 8 கடிகாரச் சுற்றுகளுக்குத் தாழ்நிலையிலும் இருக்கும். ஒரு தொடர் அலையானது, இந்தச் செயல், திரும்பத் திரும்ப நடக்கும்போது தன்னிச்சையாகக் கிடைக்கும்.

திட்டநிரல் :

நினைவக முகவரி	இயந்திரக் குறியீடுகள்	செயற் குறிப்புகள்	இயமிகள்	விளக்கம்
FC00	3E, 96	MVI	A,96 H	முறை 3க்காக எண்ணி 2 ஐ ஆரம்ப நிலைக்குக் கொண்டு வருவதற்குக் கட்டுப்பாட்டுச் சொல்லை வாங்கு.
FC02	D3, 13	OUT	13	கட்டுப்பாட்டுச் சொல் பதிவேட்டில் எழுது.
FC04	3E, 10	MVI	A,10 H	ஈரடி எண்ணிக்கைக் காகக் கணக்கீடைப் பதினாரடி எண்ணில் வாங்கு.
FC06	D3, 12	OUT	12	எண்ணி 2 ஐ கணக்கீடு கொண்டு சுமையேற்று.
FC08	76	HLT		நிறுத்து.

உதாரணம் : 2

8253இல் ஒரு எண்ணியைக் கொண்டு சதுர அலையை உருவாக்கலாம். N=8 எனில், சதுர அலை.

எண்ணி – 2 யை, முறை – 3இன் மூலம் சதுரலை உருவாக்கப் பயன்படுத்தலாம். எண்ணிக்கை ஈரடியில் நடைபெறும்.

எண்ணி–1, முறை – 2ன் மூலம் தகைவேக மின்னாக்கியாகப் பயன்படுகிறது. கட்டுப்பாட்டுச் சொல்லைக் கீழ்க்கண்டவாறு உருவாக்கலாம்.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	1	0	1	0	1	0	0=54 H

எண்ணி 1ஐ தேர்வு செய்ய, D_7 மற்றும் D_6ஐ 01 என அமைக்க வேண்டும்.

கணக்கீட்டின் LSBஐ சுமையேற்ற, D_5 மற்றும் D_4ஐ, 01 என அமைக்கவும்.

முறை 2 க்காக, D_3, D_2, D_1 ஐ 010 என அமைக்க வேண்டும்.

ஈரடி எண்ணிக்கைக்காக D_0 ஐ, 0 வாக அமைக்கவும்.

<u>திட்ட நிரல்</u> :

நினைவக முகவரி	இயந்திரக் குறியீடுகள்	செயற் குறிப்புகள்	இயமிகள்	விளக்கம்
FC00	3E, 96	MVI	A, 96 H	முறை 3இல் எண்ணி 2ஐ தொடக்க நிலைக்குக் கொண்டு வர, கட்டுப்பாட்டுச் சொல்லை வாங்கு.
FC02	D3, 13	OUT	13	கட்டுப்பாட்டுச் சொல் பதிவேட்டில் எழுது.
FC04	3E, 08	MVI	A,08 H	N=08 என்ற கணக்கீடை வாங்கு
FC06	D3,12	OUT	12	எண்ணி 2 ஐ கணக்கீடு கொண்டு சுமையேற்று.
FC08	3E, 54	MVI	A,54 H	முறை 2இல் எண்ணி 1 ஐ தொடக்க நிலைக்குக் கொண்டுவரக் கட்டுப்பாட்டுச் சொல்லை வாங்கு.
FC0A	D3, 13	OUT	13	கட்டுப்பாட்டுச் சொல் பதிவேட்டைச் சுமையேற்று.
FC0C	3E, 06	MVI	A,06 H	N=06 என்ற கணக்கீடை வாங்கு.
FC0E	D3, 11	OUT	11	எண்ணி 1 ஐ கணக்கீடு கொண்டு சுமையேற்று.
FC10	76	HLT		நிறுத்து.

எண்ணி 2இன் வெளியீடு 4 கடிகாரத் துடிப்புகளுக்கு உயர்நிலையிலேயும், அடுத்த 4 கடிகாரத் துடிப்புக்ளுக்குத் தாழ்நிலைக்கும் செல்லும். இந்தச் செயல் தொடர்ந்து மீண்டும் மீண்டும் நடைபெறும்போது, தொடர்ச்சியான சதுரலை கிடைக்கும். எண்ணி 1இன் வெளியீடு, 5 கடிகாரத் துடிப்புகளுக்கு உயர்நிலையிலேயும், பின்னர் ஒரு கடிகார நேரத்திற்குத் தாழ்நிலைக்கும் செல்லும். இந்தச் செயல் திரும்பத் திரும்ப நடைபெறும்.

உதாரணம் : 3

சதுரலையியற்றியை 8253ஐ பயன்படுத்தி உருவாக்கலாம். N=10 தசம எண். கட்டுப்பாட்டுச் சொல்லைக் கீழ்க்கண்டவாறு உருவாக்கலாம்.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	1	1	1=97 H

எண்ணி 2 ஐ தேர்வு செய்ய D_7 மற்றும் D_6 இவைகளை 10 என அமைக்க வேண்டும்.

கணக்கீட்டின் LSBஐ சுமையேற்ற D_5 மற்றும் D_4 முறையே 01 ஆக்க வேண்டும்.

முறை 3க்காக D_3, D_2, D_1 இவற்றை 011 என வைக்க வேண்டும்.

BCD எண்ணிக்கையிற்காக D_0 வை 1 என ஆக்க வேண்டும்.

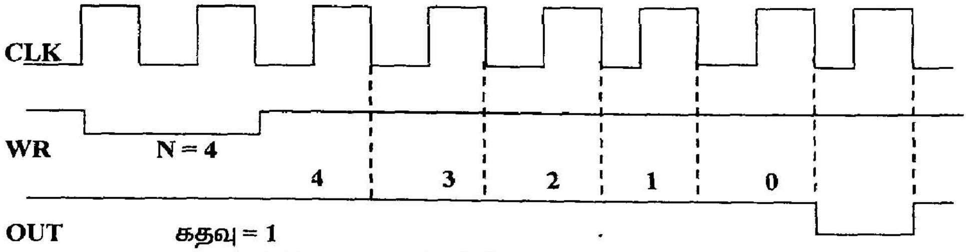
5 கடிகாரத் துடிப்புகளுக்கு வெளியீடு உயர்நிலையிலும், பின்னர் அடுத்த 5 கடிகாரத் துடிப்புகளுக்குத் தாழ்நிலையிலும் இருக்கும். N=11 எனில், 6 கடிகாரத் துடிப்புகளுக்கு உயர்நிலையிலும், அடுத்த 5 கடிகாரத் துடிப்புகளுக்குத் தாழ்நிலையிலும் இருக்கும். நேரச்சுற்றால் இந்தச் செயல் திரும்பத் திரும்ப நடைபெறுவதால், தொடர்ச்சியான சதுரலை கிடைக்கும்.

திட்டநிரல் :

பிணைவக முகவரி	இயந்திரக் குறியீடுகள்	சுஷசயற் குறிப்புகள்	இயமிகள்	விளக்கம்
FC00	3E, 97	MVI	A,97 H	BCD எண்ணிக்கை, முறை 3க்காக கட்டுப்பாட்டுச் சொல்லை வாங்கு.
FC02	D3, 13	OUT	13	கட்டுப்பாட்டுச் சொல் பதிவேட்டைச் சுமையேற்று.
FC04	3E, 10	MVI	A, 10	BCD எண்ணிக்கையிற்காகக் கணக்கீடு 10 ஐ வாங்கு. அது தசம எண்ணில் எடுக்க வேண்டும்.
FC06	D3, 12	OUT	12	எண்ணி 2ஐ கணக்கீடு கொண்டு சுமையேற்று.
FC08	76	HLT		நிறுத்து.

முறை : 4 மென்பொருள் தொடக்கித் திரிபலை (Software Triggered Strobe)

முறை 4 செயற்பாட்டின்போது, எண்ணியின் வெளியீடு ஆரம்பத்தில் உயர்நிலையில் இருக்கும். இந்த வகையில் கதவினை உயர்நிலையில் வைக்க வேண்டும். எண்ணிப் பதிவேட்டில் கணக்கீட்டைச் சுமையேற்றியவுடன், எண்ணி எண்ணிக்கையினை உடனே தொடங்கும். எண்ணி இறுதி எண்ணைத் தொட்டவுடன், ஒரு கடிகாரத் துடிப்பிற்கு வெளியீடு தாழ்நிலைக்குச் சென்று, பின்னர் உயர்நிலையை அடையும் என்பது படம் 3.13இல் காட்டப்பட்டுள்ளது.



படம். 3.13 முறை 4 மென்பொருள் தொடக்கித் திரிபலை

உதாரணம் :

8253 ஐ முறை 4இல் பயன்படுத்தி, அதன் வெளியீடு அலையை CROஇல் காணலாம்.

முறை 4இன் செயற்பாட்டிற்காக எண்ணி 1இன் கட்டுப்பாட்டுச் சொல்லைக் கீழே உள்ளவாறு உருவாக்கலாம்.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	1	1	1	1	0	0	0=78 H

எண்ணி 1 ஐ தேர்வு செய்ய, D_7 மற்றும் D_6 இவற்றை 01இல் அமைக்க வேண்டும்.

கணக்கீட்டின் LSBஐ முதலில் சுமையேற்றவும். பின்னர் MSBஐ சுமையேற்றவும் D_5 மற்றும் D_4 ஐ, 11 என வைக்க வேண்டும்.

முறை 4க்காக D_3, D_2 மற்றும் D_1 ஐ 100 என அமைக்கவும்.

ஈரடி எண்ணிக்கையிற்காக D_0 என வைக்கவும்.

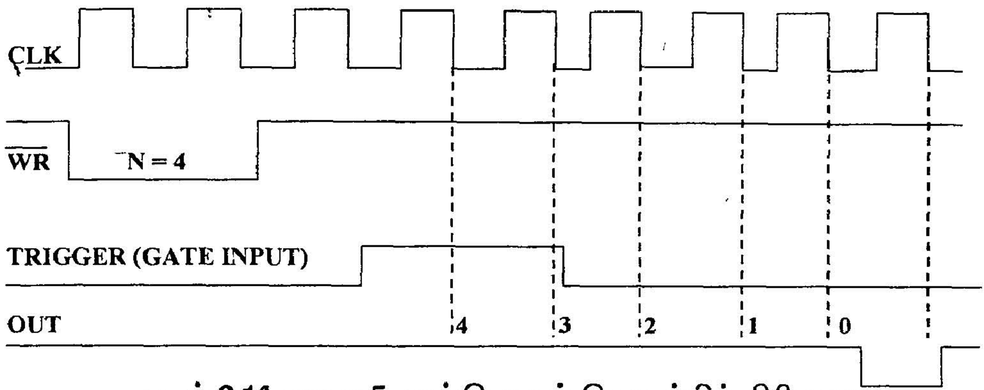
கதவை உயர்நிலையில் வைக்கவேண்டும். CRO இன் மூலம் பார்க்கும்போது, ஒரு கடிகாரத் துடிப்பிற்கு வெளியீடு தாழ்நிலைக்குச் செல்லும். எண்ணிக்கையை மிகவும் தாழ்நிலையில் வைக்க வேண்டும். ஏனெனில், திட்டநிரல் எண்ணிக்கையை மறுசுமையேற்றுவதற்கு முன்பாகவே, எண்ணிக்கை முடிந்து விடும். எண்ணிக்கை போதியளவு அதிகமாக இருந்தால், நேரத் தாமதத்தைத் திட்டநிரலானது நினைவக இருப்பிடம் FC0C லிருந்து FC04 க்குத் தாவுவதற்கு முன்பாகவே புகுத்த வேண்டும். திட்டநிரலில் வளைச்சுற்றைப் புகுத்தும்போது, திரிபலை சைகை மீண்டும் மீண்டும் உருவாதலால், இதனை CRO-இல் காண முடியும்.

திட்டநிரல் :

பிணைவக முகவரி	இயந்திரக் குறியீடுகள்	குறிப் பீடுகள்	கூடிசயற் குறிப்புகள்	இயமிகள்	விளக்கம்
FC00	**3E, 78**		**MVI**	**A,78 H**	முறை எண்ணி 1 ஐ ஆரம்ப நிலைக்குக் கொண்டு வரக் கட்டுப்பாட்டுச் சொல்லை வாங்கு.
FC02	**D3, 13**		**OUT**	**13**	கட்டுப்பாட்டுச் சொல்லைப் பதிவேட்டில் எழுது.
FC04	**3E, 05**	**LOOP**	**MVI**	**A, 05**	கணக்கீட்டின் **LSB**ஐ வாங்கு.
FC06	**D3, 11**		**OUT**	**11**	கணக்கீட்டின் **LSB**ஐ எண்ணி 1இல் சுமையேற்று.
FC08	**3E, 00**		**MVI**	**A,00**	கணக்கீட்டின் **MSB**ஐ வாங்கு.
FC0A	**D3, 11**		**OUT**	**11**	கணக்கீட்டின் **MSB**ஐ எண்ணி 1இல் சுமையேற்று.
FC0C	**C3, 04, FC**		**JMP**	**LOOP**	

முறை : 5 வன்பொருள் தொடக்கித் திரிபுலை (Hardware Triggered Strobe)

வன்பொருள் தொடக்கித் திரிபலைச் செயற்பாட்டில், கதவு உள்ளீட்டுத் தொடக்கியாக வேலை செய்யும். இந்த வகையை அமைத்த உடன், வெளியீடு, ஆரம்பத்தில் உயர்நிலையில் இருக்கும். எண்ணிக்கை மதிப்பு Nஐ எண்ணியில் சுமையேற்ற வேண்டும். கதவு உள்ளீடு தாழ்நிலையிலிருந்து உயர்நிலைக்கு மாறும்போது, எண்ணி எண்ணிக்கையினைக் குறைக்கத் தொடங்கும். கடிகாரத்தின் முதல் எதிர் விளிம்பின்போது, எண்ணிக்கை தொடங்கும். இதனைப் படம் 3.14இல் காணலாம்.



படம் 3.14 முறை 5 வன்பொருள் தொடக்கித் திரிபலை

இறுதி எண்ணிக்கையின்போது, ஒரு கடிகார நேரத்திற்கு வெளியீடு தாழ்நிலைக்குச் சென்று, பின்னர் மீண்டும் உயர்நிலையை அடையும். கதவு உள்ளீட்டில் தாழ்நிலையிலிருந்து உயர்நிலைக்கு மாறுவது, இந்த முறையைத் தொடக்குவதற்குக் காரணமாகிறது. அதனால், இதற்கு வன்பொருள் தொடக்கி திரிபலை எனப் பெயர். முறை 1ஐ போல், இந்த முறையும், மறுத்தொடக்கியாகச் செயல்படுகிறது. கதவு உள்ளீடு தாழ்நிலையிலிருந்து உயர்நிலைக்கு மாறும்போது, எண்ணி மீண்டும் கணக்கீட்டு மதிப்பு N ஆல் மறுசுமையேற்றப்படுகிறது. எண்ணி மறுபடியும் எண்ணிக்கையில் குறையத்தொடங்கும். மேலும், இறுதி எண்ணிக்கையின்போது, வெளியீடு ஒரு கடிகார நேரத்திற்குத் தாழ்நிலைக்குச் செல்லும்.

உதாரணம் :

8253 இல் முறை 5 ஐ பயன்படுத்தி CRO இல் வெளியீட்டின் அலை உருவத்தைக் காணலாம். முறை 5இல் எண்ணி 1 ற்கான கட்டுப்பாட்டுச் சொல் கீழே கொடுக்கப்பட்டுள்ளது.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	1	1	1	1	0	1	0=7A H

எண்ணி 1 ஐ தேர்வு செய்ய, D_7 மற்றும் D_6 இவற்றை 01 என அமைக்கவும்.

எண்ணிக்கையின் **LSB** ஐ முதலிலும், பின்னர் **MSB** ஐயுப் சுமையேற்ற D_5 மற்றும் D_4 முறையே 11 என இருக்க வேண்டும்.

முறை 5 ற்காக D_3, D_2, D_1, என்பனவற்றை, முறையே 101=5 என வைக்க வேண்டும்.

ஈரடி எண்ணிக்கைக்கு D_0 வை 0 வில் அமைக்க வேண்டும்.

8255.2 இன் **PC0** முனை கதவுடன் இணைக்கப்பட்டுள்ளது. நுண்முறைப்படுத்தியினால் உண்டாக்கப்பட்ட துடிப்பு, **PC0**இல் பெறக்கூடியது. இந்தத் துடிப்பு, கதவிற்குக் கொடுக்கப்படும். செயல்வகை 5ல் கதவு உள்ளீட்டின் தாழ்நிலையிலிருந்து உயர்நிலை மாற்றத்தின்போது எண்ணிக்கையானது தொடங்கும். ஆகவே, துடிப்புச் சைகையைக் கதவிற்குக் கொடுக்க வேண்டும். எண்ணிக்கையை முதலில் குறைந்த மதிப்பில் வைக்கவேண்டும். ஏனெனில், கதவிற்கு அடுத்த துடிப்பைக் கொடுப்பதற்கு முன்பாகவே, எண்ணிக்கை முடிந்துவிடும். எண்ணிக்கையைப் போதியளவு பெரியதாக வைத்தால், திட்ட நிரலானது **FC18**லிருந்து **FC0C**க்குத் தாவுவதற்கு முன் தாமத நேரத்தைப் புகுத்துகிறது. வெளியீட்டு அலையின் உருவமைப்பு மற்றும் கடிகாரத்தை, **CRO**இல் காணலாம்.

திட்ட நிரல் :

நினைவக முகவரி	எந்திரக் குறியீடுகள்	குறிப் பீடுகள்	செயற் குறிப்புகள்	இயமிகள்	விளக்கம்
FC00	3E,7A		MVI	A,7A	முறை 5இல் எண்ணி 1ஐ தொடக்க நிலையில் வைக்கக் கட்டுப்பாட்டுச் சொல்லை வாங்கு.
FC02	DE, 13		OUT	13	கட்டுப்பாட்டுச் சொல் பதிவேட்டில் எழுது.
FC04	3E, 08		MVI	A, 08	எண்ணிக்கையின் LSBஐ வாங்கு.
FC06	D3, 11		OUT	11	எண்ணி 1 ஐ எண்ணிக்கையின் LSB கொண்டு சுமையேற்று.
FC08	3E, 00		MVI	A,00	எண்ணிக்கையின் MSBஐ வாங்கு.
FC0A	D3, 11		OUT	11	எண்ணி 1 ஐ எண்ணிக்கையின் MSB கொண்டு சுமையேற்று.
FC0C	3E, 80	LOOP	MVI	A, 80H	8255. 2இன் துறைமுகங்களைத் தொடக்க நிலையில் வை.
FC0E	D3, 0B		OUT	0B	
FC10 FC12	3E, 00 D3, 00		MVI OUT	A,00 OA	PCOஇல் வெளியிடக் கூடிய துடிப்பை உண்டாக்கு. அது 8253இன் கதவுடன் இணைக்கப்படவேண்டும்.
FC14	3E, 01	MVI	A, 01		
FC16	D3, 0A	OUT	0A		
FC18	C3, 0C, FC	JMP	LOOP		

4

இன்டெல் 8086 நுண்முறைப்படுத்தி

(Intel 8086 Microprocessor)

4.1 முன்னுரை

(Introduction)

இன்டெல் 8086 ஒரு 16-பிட் நுண்முறைப்படுத்தியாகும். இதை நுண்கணினியின் மையச்செயலகமாகப் பயன்படுத்தலாம். 16-பிட் என்ற சொல் இன்டெல் 8086இன் கணித வாதியல் அலகு, உள்ளார்ந்த பதிவேடுகள் மற்றும் பல கட்டளைகள், 16-பிட் ஈரடி வார்த்தைகளைக் கொண்டு இயங்குமாறு வடிவமைக்கப்பட்டதைக் குறிக்கிறது. இந்த நுண்முறைப்படுத்தியில், 16-பிட் தரவு ஊர்தி (Data bus) மற்றும் 20-பிட் முகவரி ஊர்தி (Address bus) ஆகியவை உள்ளன. 16-பிட் அல்லது 8-பிட் தரவுகளை, நினைவகத்திலிருந்து படிக்கவோ அல்லது நினைவகத்தில் எழுதவோ இந்த நுண்முறைப்படுத்தியால் முடியும். அதேபோன்று 2^{20} அல்லது 1,048,576 நினைவக இருப்பிடங்களில், ஏதாவது ஒரு இருப்பிடத்தைத் தேடிக் கண்டுபிடிக்க முடியும். விரிவடைந்த நினைவகத் தேடும் திறன் மற்றும் பல முக்கியமான பாகங்களால் இன்டெல் 8086 நுண்முறைப்படுத்தியின் பல்திட்ட நிரலாக்குத் திறன் அதிகரிக்கப்படுகிறது. பின்வரும் பிரிவுகளில் 8086 நுண்முறைப்படுத்தியின் கட்டமைப்பு, மின்முனை அமைப்பு மற்றும் அதன் கட்டளை வகைகளைப் பற்றிப் பார்ப்போம்.

4.2 8086இன் உள்ளார்ந்த கட்டமைப்பு

(Internal Architecture of 8086)

படம்(4.1)இல் இன்டெல் 8086இன் உள்ளார்ந்த கட்டமைப்பு காட்டப்பட்டுள்ளது. இதன் மையச்செயலகம் இரண்டு பாகங்களாகப் பிரிக்கப்படுகிறது. அவை

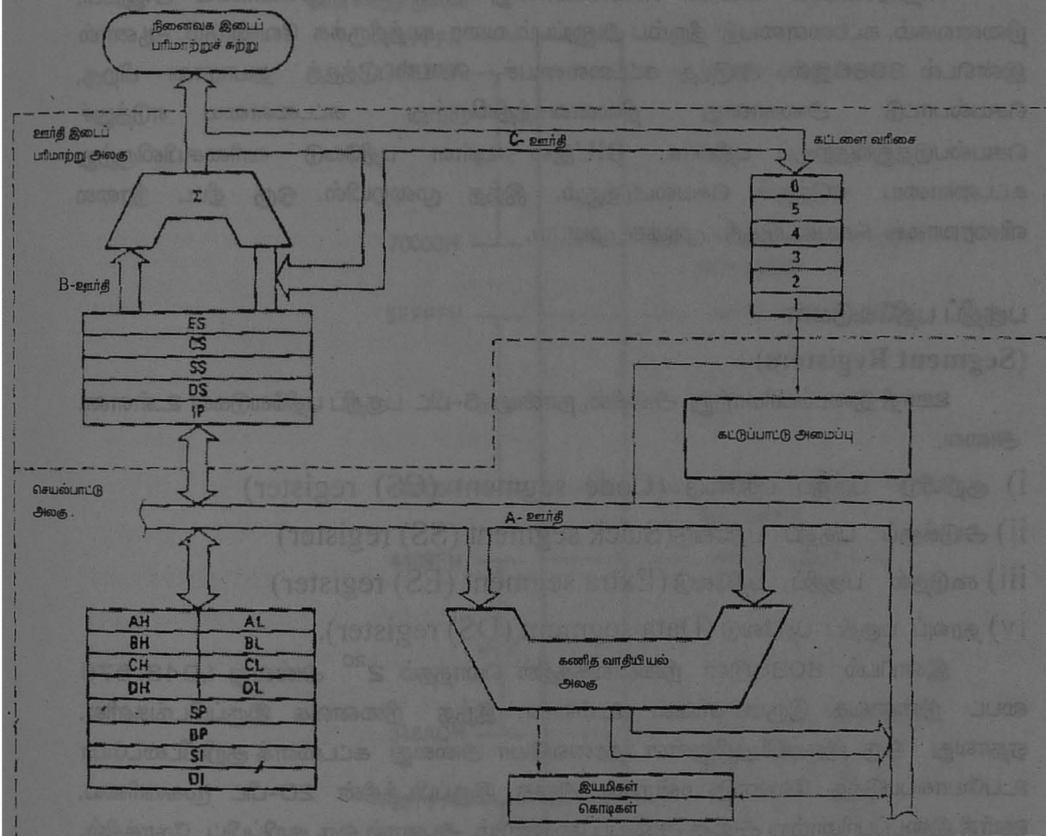
i) ஊர்தி இடைப்பரிமாற்று அலகு (Bus Interface Unit – BIU)

ii) செயல்பாட்டு அலகு (Execution Unit – EU)

4.2.1 ஊர்தி இடைப்பரிமாற்று அலகு

(Bus Interface Unit)

இந்த அலகு, நினைவகத்திலிருந்து கட்டளைகளைக் கொணர்தல், தரவுகளை நிலையங்கள் (Ports) மற்றும் நினைவகத்திலிருந்து படித்தல் மற்றும் தரவுகளை நிலையங்கள் மற்றும் நினைவகத்தில் எழுதுதல் போன்ற வேலைகளை செய்யும். அதாவது இந்த அலகு, செயல்பாட்டு அலகிற்கான எல்லாவிதமான தரவு மற்றும் முகவரி மாற்றத்தையும் கையாளுகிறது. ஊர்தி இடைப்பரிமாற்று அலகின் முக்கியமான பாகங்களைப் பற்றி பின்வரும் பிரிவுகளில் விரிவாகப் பார்ப்போம்.



படம் (4.1) 8086 இன் உள்ளார்ந்த கட்டமைப்பு

கட்டளை வரிசைப் பதிவேடு
(Instruction Queue Register)

ஒரு திட்டநிரலைச் செயல்படுத்துவதை வேகப்படுத்துவதற்காக, BIU, ஆறு கட்டளைகளை நினைவகத்திலிருந்து முன்னதாகவே கொணரும். இந்தக் கட்டளைகள், செயல்பாட்டு அலகிற்ககாகப் (EU) பதிவேடுகளின் தொகுப்புகளில் வைக்கப்பட்டிருக்கும். இந்தப் பதிவேடுகளின் தொகுப்பே, கட்டளை வரிசைப் பதிவேடு (Instruction Queue Register) எனப்படும். கட்டளைகளைக் கொணர்தல் வேலையை, ஊர்தி இடைப்பரிமாற்று அலகும் (BIU), அதைச் செயல்படுத்தும் வேலையைச் செயல்பாட்டு அலகும் (EU) செய்கின்றன.

சாதாரணமாக மையச் செயலகமானது நினைவக முகவரியை அனுப்பி, நினைவகம், கட்டளையைத் திரும்ப அனுப்பும் வரை காத்திருக்க வேண்டும். ஆனால் இன்டெல் 8086இல், அடுத்த கட்டளையைச் செயல்படுத்தத் தயாரான பிறகு, செயல்பாட்டு அலகானது, நினைவகத்திலிருந்து கட்டளையை எடுத்துச் செயல்படுத்துவதற்குப் பதிலாக, BIUஇல் உள்ள பதிவேடு வரிசையிலிருந்து கட்டளையை எடுத்துச் செயல்படுத்தும். இந்த முறையில், ஒரு திட்ட நிரலை விரைவாகச் செயல்படுத்தி முடிக்க முடியும்.

பகுதிப் பதிவேடுகள்
(Segment Registers)

ஊர்தி இடைப்பரிமாற்று அலகில், நான்கு 6-பிட் பகுதிப் பதிவேடுகள் உள்ளன. அவை,

i) குறியீடு பகுதிப் பதிவேடு (Code segment (CS) register)

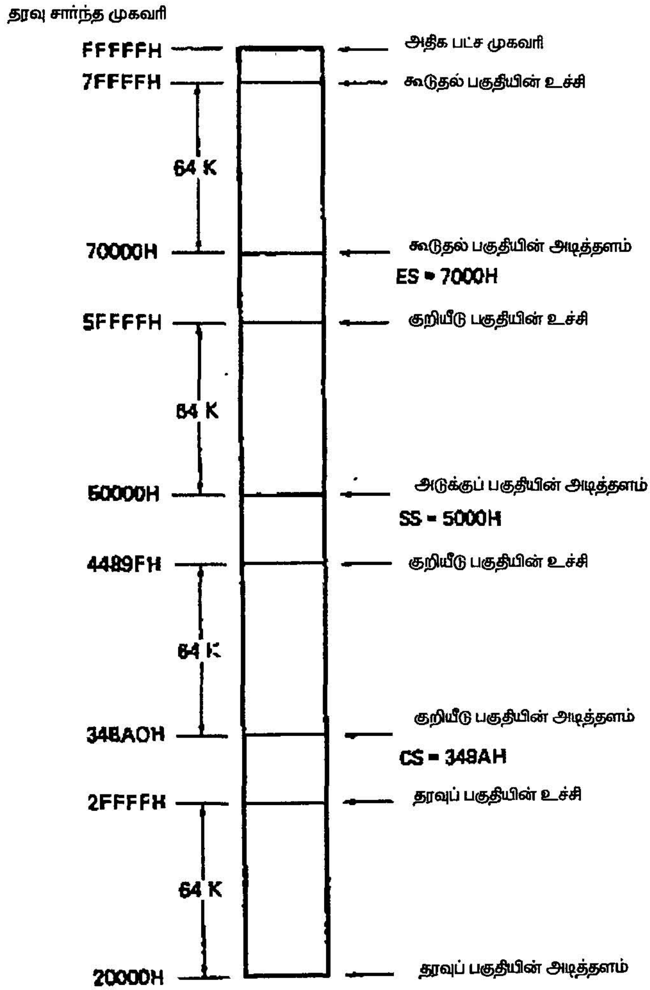
ii) அடுக்குப் பகுதிப் பதிவேடு(Stack segment (SS) register)

iii) கூடுதல் பகுதிப் பதிவேடு (Extra segment (ES) register)

iv) தரவுப் பகுதிப் பதிவேடு (Data segment (DS) register).

இன்டெல் 8086இன் நினைவகத்தில் மொத்தம் 2^{20} அல்லது 1,048, 576 பைட் நினைவக இருப்பிடங்கள் உள்ளன. இந்த நினைவக இருப்பிடங்களில், ஏதாவது ஒரு இருப்பிடத்திலுள்ள தரவையோ அல்லது கட்டளைக் குறியீட்டையோ உபயோகப்படுத்த வேண்டுமென்றால், அந்த இருப்பிடத்தின் 20-பிட் முகவரியை, ஊர்தி இடைப்பரிமாற்று அலகு அனுப்ப வேண்டும். ஆனால் ஒரு குறிப்பிட்ட நேரத்தில், 8086இன் ஊர்தி இடைப்பரிமாற்று அலகானது, 1 மெகா பைட்டுகளில், 65, 536 பைட் பகுதிகளை மட்டும் பயன்படுத்தும். ஒரு குறிப்பிட்ட நேரத்தில், நான்கு நினைவகப் பகுதிகளில் எந்த ஒரு நினைவகப் பகுதியைக் கொண்டு இன்டெல் 8086 வேலை செய்கிறதோ, அந்த நினைவகப் பகுதியின் தொடக்க முகவரியில் முதல் 16-பிட்டுகள் இந்தப் பகுதிப் பதிவேடுகளில் இருக்கும்.

8086இன் 1 மெகா பைட் நினைவகம், எவ்வாறு நான்கு 64கிலோ பைட் பகுதிகளாகப் பிரிக்கப்படுகிறது என்பதைப் படம் (4.2) தெளிவாக விளக்குகிறது.



படம் (4.2) 1 மெகா பைட் நினைவகம், எவ்வாறு நான்கு 64K பைட் பகுதிகளாகப் பிரிக்கப்படுகிறது என்பதைக் காட்டுகிறது.

குறியீடு பகுதிப் பதிவேடு
(Code segment (CS) register)

நினைவகத்தின் இந்தப் பகுதியில், திட்டநிரலின் கட்டளைகள் சேமித்து வைக்கப்பட்டிருக்கும். ஊர்தி இடைப்பரிமாற்று அலகு, எந்தப் பகுதியிலிருந்து கட்டளைக் குறியீடு பைட்டுகளைக் கொணருகிறதோ, அந்தப் பகுதியின் தொடக்க முகவரியின் முதல் 16-பிட்டுகள், குறியீடு பகுதிப் பதிவேட்டில் இருக்கும். மீதமுள்ள நான்கு பிட்டிற்குப் பதிலாக BIU சுழியைச் சேர்க்கும். உதாரணத்திற்கு, CS பதிவேட்டில் 348AH என்ற 16-பிட் முகவரி இருந்தால், CS இன் தொடக்க முகவரி 348A0H ஆக இருக்கும். பகுதிப் பதிவேட்டில் (Segment register) சேமித்து வைக்கப்பட்டிருக்கும் தொடக்க முகவரியின் ஒரு பகுதி, பகுதி அடித்தளம் (Segment base) எனப்படும்.

அடுக்குப் பகுதிப் பதிவேடு
(Stack Segment Register)

துணைத் திட்டநிரலைச் செயல்படுத்தும்போது முகவரி மற்றும் தரவுகளைச் சேமித்து வைப்பதற்காக ஒதுக்கப்பட்ட நினைவகத்தின் ஒரு பகுதியே, அடுக்கு எனப்படும். அடுக்குப் பகுதியின் தொடக்க முகவரியில் முதல் 16-பிட்டுகள், அடுக்குப் பகுதி பதிவேட்டில் சேமித்து வைக்கப்படும்.

தரவுப் பகுதிப் பதிவேடு
(Data Segment register)

தரவுப் பகுதியின் தொடக்க முகவரியின் முதல் 16-பிட்டுகள், தரவுப் பகுதிப் பதிவேட்டில் (Data Segment (DS) register) சேமித்து வைக்கப்படும். தரவுப் பகுதியில் தரவுகள் சேமித்து வைக்கப்படும். மூலச் சுட்டிக்காட்டிப் பதிவேடு (Source index (SI) register) அல்லது சேர்ப்பிடச் சுட்டிக்காட்டிப் பதிவேடு (Destination index (DI) register) ஆகியவற்றின் உள்ளடக்கம், DS பதிவேட்டின் உள்ளடக்கத்தோடு சேர்ந்து 20-பிட் தரவு சார்ந்த முகவரி (Physical address) கிடைக்கும். இந்த முகவரி, மூல மற்றும் சேர்ப்பிட இயமியின் முகவரியாகும். அதாவது SI அல்லது DIல் செயலூக்கமுடைய முகவரி (effective address) இருக்கும்.

கூடுதல் பகுதிப் பதிவேடு
(Extra segment register)

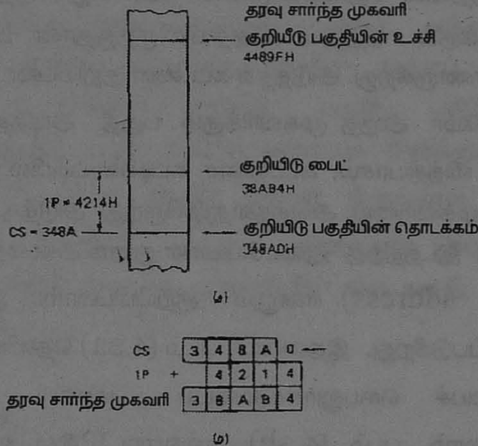
இந்தப் பகுதியும், தரவுகளைச் சேமித்து வைக்கப் பயன்படுகிறது. இந்தப் பகுதியின் தொடக்க முகவரியின் முதல் 16-பிட்டுகள், கூடுதல் பகுதிப் பதிவேட்டில் இருக்கும்.

கட்டளைச் சுட்டிக்காட்டி
(Instruction Pointer)

குறியீடு பகுதியின் தொடக்க முகவரி அல்லது பகுதி அடித்தள முகவரி, CS பதிவேட்டில் இருக்கும். இந்தப் பகுதியிலிருந்துதான் BIU, கட்டளைக் குறியீடு பைட்டுகளைக் கொணருகிறது. அடுத்தக் கட்டளை குறியீட்டை, எந்த முகவரியிலிருந்து கொணர வேண்டுமோ அந்த முகவரிக்கும் பகுதி அடித்தள (Segment base) முகவரிக்குமுள்ள வித்தியாசம், கட்டளைச் சுட்டிக்காட்டியில் (Instruction Pointer) இருக்கும். இந்த முகவரியை, அடித்தளத்திலிருந்து அடுத்த கட்டளைக் குறியீட்டைக் கொணர வேண்டிய இடத்திற்கு இடையேயான தூரம் அல்லது இடப்பெயர்ச்சி முகவரி (Displacement address) என்றும் குறிப்பிடலாம். இந்தத் தூரமே IPஇல் சேமித்து வைக்கப்படுகிறது. இதனைப் படம் (4.3a) தெளிவாக காட்டுகிறது. IPஇல் உள்ள முகவரியைச் செயலூாக்கமுடைய முகவரி (Effective address) எனவும் குறிப்பிடலாம். படம் (4.3b), எவ்வாறு IPஇல் உள்ள 16-பிட் முகவரி CSஇல் உள்ள 16-பிட் முகவரியுடன் கூட்டப்பட்டு, அடுத்த கட்டளைக் குறியீடு இருக்கக்கூடிய இருப்பிடத்தின் தரவு சார்ந்த 20-பிட் முகவரி பெறப்படுகிறது என்பதைத் தெளிவாக விளக்குகிறது. இரண்டு 16-பிட் முகவரிகளும் நேரடியாகக் கூட்டப்படாமல், CSஇல் உள்ள 16-பிட் முகவரி, இடது பக்கமாக நான்கு பிட்டுகள் நகர்த்தப்படுகிறது. இவ்வாறு நகர்த்துவதால், உண்மையான 20-பிட் பகுதி அடித்தள முகவரி கிடைக்கிறது. பிறகு இந்த முகவரி, IPஇல் உள்ள முகவரியுடன் கூட்டப்பட்டுத் தரவு சார்ந்த 20-பிட் முகவரி கிடைக்கிறது. உதாரணத்திற்கு CSஇல் 348AH என்ற முகவரி இருப்பதாகக் கருதுவோம். இதை இடதுபக்கமாக நான்கு பிட்டுகள் நகர்த்தினால், 348A0H என்ற 20-பிட் முகவரி கிடைக்கும். இது குறியீடு பகுதியின் தொடக்க முகவரியாகும். இப்பொழுது IPஇல் 4214H என்ற இடப்பெயர்ச்சி முகவரி இருப்பதாகக் கருதுவோம். இந்த முகவரியை மேலே சொல்லப்பட்ட 20-பிட் முகவரியுடன் கூட்டினால், 38AB4H என்ற தரவு சார்ந்த முகவரி (Physical address) கிடைக்கும். இது 20-பிட் ஆக இருக்கும். நினைவகத்தின் குறியீடு பகுதியிலுள்ள தரவு சார்ந்த முகவரியைக் கணக்கீடு செய்ய CS மற்றும் IP பதிவேடுகள் பயன்படுத்தப்படுகின்றன. இந்த முறையில், நினைவகத்தின் மற்ற பகுதியிலுள்ள தரவு சார்ந்த முகவரியையும் கணக்கீடு செய்யலாம். தரவு சார்ந்த முகவரியைக் கணக்கீடு செய்யப் பயன்படுத்தப்படும் சமன்பாடு, கீழே கொடுக்கப்பட்டுள்ளது.

தரவு சார்ந்த முகவரி = பகுதி அடித்தள முகவரி + செயலூாக்கமுடைய முகவரி
(Physical address) = (segment base address) + (effective address)

ஒவ்வொரு பகுதியிலும், எவ்வாறு தரவு சார்ந்த முகவரி கணக்கீடு செய்யப்படுகிறது என்பதைப் பார்ப்போம்.



படம் (4.3) IP மற்றும் CSஐ கூட்டிக் குறியீடு பைட்டின் முகவரியைக் கணக்கிடுதல்

குறியீடு பகுதி

(Code segment)

PA = (CS)0 + EA

செயலூாக்கமுடைய முகவரி (EA), கட்டளைச் சுட்டிக்காட்டியில் (IP) இருக்கும். CS பதிவேட்டில் உள்ள முதல் 16-பிட்டுடன் நான்கு சுழி பிட்டுகள் சேர்த்தால், குறியீடு பகுதியின் தொடக்க முகவரியான 20-பிட் முகவரி கிடைக்கும். இதை CS (0) எனக் குறிப்பிடலாம்.

தரவுப் பகுதி

(Data segment)

PA = (DS)0 + EA

செயலூாக்கமுடைய முகவரி SI அல்லது DIஇல் இருக்கும். பதிவேடு, மறைமுக முகவரியிடல் வகையில் BX பதிவேட்டில் இருக்கும்.

அடுக்குப் பகுதி

(Stack segment)

PA = (SS)0 + EA

செயலூாக்கமுடைய முகவரி SP அல்லது BPஇல் இருக்கும்.

கூடுதல் பகுதி
(Extra segment)
PA = (ES)0 + EA
செயலூக்கமுடைய முகவரி DI இல் இருக்கும்.

4.2.2 செயற்பாட்டு அலகு
(Execution Unit – EU)

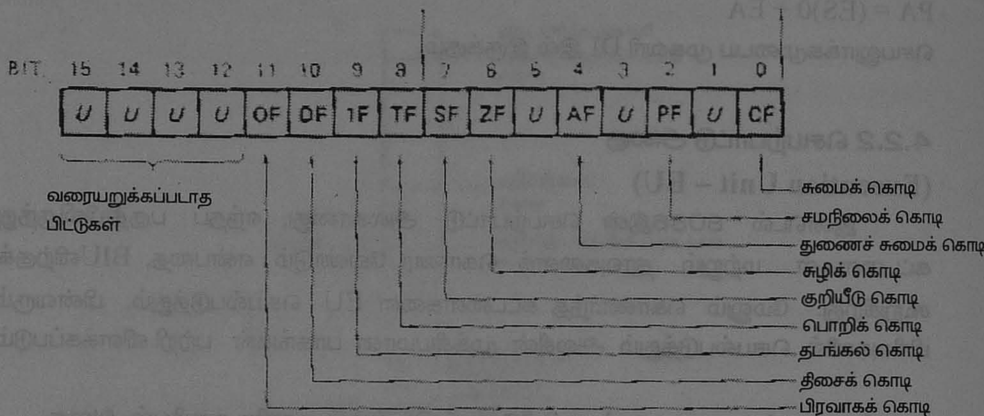
இன்டெல் 8086இன் செயற்பாட்டு அலகானது, எந்தப் பகுதியிலிருந்து கட்டளைகள் மற்றும் தரவுகளைக் கொணர வேண்டும் என்பதை, BIUவிற்குக் கூறுகிறது. மேலும் கொணர்ந்த கட்டளைகளை EU செயல்படுத்தும். பின்வரும் பிரிவுகளில், செயல்படுத்தும் அலகின் முக்கியமான பாகங்கள் பற்றி விளக்கப்படும்.

கட்டுப்பாடு சுற்று, கட்டளைக் குறியீடு மாற்றி மற்றும் கணித வாதியல் அலகு
Control Circuitry Instruction decoder and ALU

செயற்பாட்டு அலகிலுள்ள கட்டுப்படுத்தும் மின்சுற்றானது, உள்ளார்ந்த செயற்பாடுகளைக் கட்டுப்படுத்துகிறது. குறியீடு மாற்றியானது, நினைவகத்திலிருந்து கொணரப்பட்ட கட்டளைகளைச் செயல்பாட்டு அலகிற்குத் தகுந்தவாறு குறியீடு மாற்றம் செய்து, தொடர் வேலைகளாகத் தருகிறது. இந்த வேலைகளைச் செயற்பாட்டு அலகு செய்து முடிக்கும். EUவில் ஒரு 16-பிட் கணித வாதியல் அலகு உள்ளது. இந்த அலகு கூட்டல், கழித்தல், உம், அல்ல, தவிர்க்கப்பட்ட அல்ல, ஒரு பிட் குறைத்தல், ஒரு பிட் அதிகரித்தல் மற்றும் ஈரடி எண்களை நகர்த்துதல் போன்ற வேலைகளைச் செய்கிறது.

கொடிப் பதிவேடு
(Flag Register)

இது ஒரு நிலைமாற்றியாகும் (Flipflop). இந்த நிலைமாற்றியானது, செயற்பாட்டு அலகின் சில வேலைகளைக் கட்டுப்படுத்தும்போதோ அல்லது கட்டளைகளைச் செயல்படுத்தும்போதோ வரக்கூடிய சில நிபந்தனைகளை உணர்த்தும். ஒரு 16-பிட் கொடிப் பதிவேட்டில் மொத்தம் ஒன்பது கொடிகள் இருக்கும். கொடிப் பதிவேட்டில் எவ்வாறு ஒன்பது கொடிகள் இடம் பெற்றுள்ளன என்பதைப் படம் (4.4) தெளிவாகக் காட்டுகிறது. ஒன்பது கொடிகளில் ஆறு கொடிகள், கட்டளைகளைச் செயல்படுத்தும்போது வரக்கூடிய சில நிபந்தனைகளை உணர்த்துகின்றன. ஆறு கொடிகளாவன i) சுமைக் கொடி ii) சமநிலைக் கொடி iii) துணைச் சுமைக் கொடி iv) சுழிக் கொடி v) குறியீடு கொடி மற்றும் vi) பிரவாகக் கொடி.



படம் (4.4) 8086 இன் கொடி பதிவேடு அமைப்பு

சுமைக் கொடி
(CARRY FLAG)

இரண்டு 16-பிட் ஈரடி எண்களைக் கூட்டும்போது சுமை வந்தால், சுமைக் கொடியானது ஒன்று என அமைவு பெறும். சுமை வராவிட்டால் சுழி என்று அமைவு பெறும். இந்தக் கொடியானது, முந்தைய முடிவு, எதிர்க் குறியுடன் உள்ளதா அல்லது நேர்க்குறியுடன் உள்ளதா என்பதை உணர்த்தும்.

குறியீடு கொடி
(SIGN FLAG)

இதன் மதிப்பு, ஒரு முடிவின் பெரும முக்கியத்துவம் வாய்ந்த பிட்டின் மதிப்பிற்குச் சமமாக இருக்கும்.

சுழிக் கொடி
(ZERO FLAG)

முடிவு சுழியாக இருந்தால், இந்தக் கொடி ஒன்று என அமைவு பெறும். முடிவு ஒன்றாக இருந்தால், இந்தக் கொடி சுழி என அமைவு பெறும்.

சமநிலைக் கொடி
(PARITY FLAG)

முடிவில் சிறும முக்கியத்துவம் வாய்ந்த பிட்டுகளில் (LSB), இரட்டைப் பட ஒன்றுகள் இருந்தால், இந்தக் கொடி ஒன்று என அமைவு பெறும். அவ்வாறு இல்லாவிட்டால் சுழி என அமைவு பெறும்.

துணைச் சுமைக் கொடி

(AUXCILLARY CARRY FLAG)

கூட்டலின்போது மூன்றாவது பிட்டில் சுமை வந்தாலோ, அல்லது கழித்தலின் போது மூன்றாவது பிட் கடன் வாங்கினாலோ, இந்தக் கொடி ஒன்று என அமைவு பெறும்.

பிரவாகக் கொடி

(OVER FLOW FLAG)

கூட்டலின்போது பெரும முக்கியத்துவம் வாய்ந்த பிட்டில் சுமை சேர்ந்தாலோ, அல்லது அந்த பிட்டில் சுமை ஏற்படாவிட்டாலோ, இந்தக் கொடி ஒன்று என அமைவு பெறும். கழித்தலின்போது பெரும முக்கியத்துவம் வாயந்த பிட்டிற்குக் கடன் தேவைப்பட்டாலோ, அல்லது இந்தப் பிட்டிலிருந்து கடன் பெறப்படாவிட்டாலோ, பிரவாகக் கொடி ஒன்று என அமைவு பெறும்.

மேற்கூறப்பட்ட கொடிகளின் செயலை, ஒரு சிறிய உதாரணம் மூலம் விளக்கலாம். முந்தைய கட்டளை, ஒரு கூட்டல் செயலைச் செயல்படுத்துவதாகக் கருதுவோம்.

$$\begin{array}{r} 0010\ 0011\ 0100\ 0101 \\ \underline{0011\ 0010\ 0001\ 1001} \\ 0101\ 0101\ 0101\ 1110 \end{array}$$

பிறகு

SF = 0 ZF = 0 PF = 0 CF = 0 AF = 0 OF = 0

கொடிப் பதிவேட்டில் உள்ள மீதமுள்ள மூன்று கொடிகள், நுண்முறைப்படுத்தியின் சில வேலைகளைக் கட்டுப்படுத்த உதவுகின்றன. அவை, i) பொறிக் கொடி ii) தடங்கல் கொடி iii) திசைக் கொடி

பொறிக் கொடி

(Trap Flag)

இந்தக் கொடி அமைவு பெற்றால், 8086 நுண்முறைப்படுத்தி, ஒரு நேரத்தில் ஒரு கட்டளையை மட்டும் செயல்படுத்தும்.

தடங்கல் கொடி

(Interrupt Flag)

இந்தக் கொடி ஒன்று என அமைவு பெற்றால், தடங்கல் சைகைகள் செயலியக்கம் பெறும். சுழி என்று அமைவு பெற்றால், எல்லாத் தடங்கல் சைகைகளும் செயல் முடக்கப்படும்.

திசைக் கொடி
(Direction Flag)

இந்தக் கொடி ஒன்று என அமைவு பெற்றால் கம்பிக் கட்டளை (String Instruction), தன்னிச்சையாக முகவரியைக் குறைக்கும். சுழி என அமைவு பெற்றால், கம்பி முகவரி (String Address) அதிகரிக்கப்பட வேண்டும்.

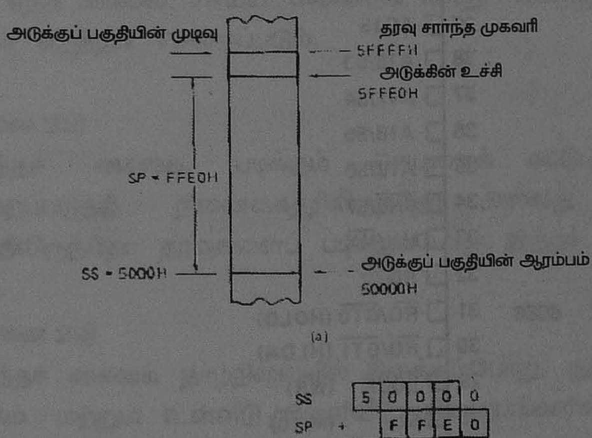
பொது நோக்கப் பதிவேடுகள்
(General Purpose Register)

இன்டெல் 8086இன் செயற்பாட்டு அலகில், மொத்தம் எட்டு பொது நோக்கப் பதிவேடுகள் உள்ளன. அவை AH, AL, BH, BL, CH, CL, DH மற்றும் DL ஆகும். இதில் AL பதிவேடு, சேமக்கலம் என்று அழைக்கப்படுகிறது. இந்தப் பதிவேடுகள், 8-பிட் தரவுகளைத் தற்காலிகமாகச் சேமித்து வைக்கப் பயன்படுகிறது. பொது நோக்கப் பதிவேடுகளில் சில, ஜோடியாகச் சேர்ந்து 16-பிட் தரவுகளைச் சேமிக்க உதவுகின்றன. AH மற்றும் AL, BH மற்றும் BL, CH மற்றும் CL, DH மற்றும் DL ஆகியவை, அனுமதிக்கப்பட்ட பதிவேடு ஜோடிகளாகும். AH – AL ஜோடி AX பதிவேடு என்றும், BH – BL ஜோடி BX பதிவேடு என்றும், CH – CL ஜோடி CX பதிவேடு என்றும், DH – DL ஜோடி DX பதிவேடு என்றும் குறிப்பிடப்படுகின்றன. 16-பிட் வேலைகளுக்கு AX பதிவேடு சேமக்கலமாக செயல்படுகிறது.

அடுக்குச் சுட்டிக்காட்டிப் பதிவேடு
(Stack Pointer Register)

துணைத் திட்டநிரல் செயல்படுத்தும்போது, நினைவகத்தின் ஒரு பிரிவு, முகவரி மற்றும் தரவுகளைச் சேமித்து வைப்பதற்காகப் பயன்படுத்தப்படும். இதை அடுக்கு எனலாம். 8086இல் நினைவகத்தின் ஒரு முழு 64Kபைட் பகுதி அடுக்காகச் செயல்படுத்த அனுமதிக்கப்படுகிறது. இந்தப் பகுதியின் தொடக்க முகவரி, அடுக்குப் பகுதி பதிவேட்டில் (Stack Segment Register) வைக்கப்படும். அடுக்கில், எந்த ஒரு நினைவக இருப்பிடத்தில் ஒரு வார்த்தை சேமிக்கப்பட்டதோ,அந்த முகவரிக்கும், அடுக்குப் பகுதியின் தொடக்க முகவரிக்கும் இடையேயுள்ள முகவரி வித்தியாசம், அடுக்குச் சுட்டிக்காட்டியில் சேமிக்கப்படும். எந்த நினைவக இருப்பிடத்தில் கடைசியாக பைட்டுகள்(வார்த்தை)சேமிக்கப்பட்டனவோ அந்த நினைவக இருப்பிடத்தின் முகவரி, அடுக்கு உச்சி (Stack Top) எனப்படும். இவை படம் (4.5a)இல் தெளிவாகக் காட்டப்பட்டுள்ளன.

அடுக்கைப் படித்தல் மற்றும் அடுக்கில் எழுதுதல் போன்ற வேலைகளைச் செய்யத் தேவையான முகவரி, பின்வருமாறு பெறப்படுகிறது. அடுக்குச் சுட்டிக்காட்டிப் பதிவேட்டில் உள்ளதை, அடுக்குப் பகுதிப் பதிவேட்டில் உள்ள முகவரியுடன் கூட்ட வேண்டும். இதைச் செய்வதற்கு, அடுக்குப் பகுதிப் பதிவேட்டில் உள்ள முகவரியை இடது பக்கமாக நான்கு பிட் நகர்த்த வேண்டும். பிறகு அடுக்குசுட்டிக்காட்டியில் உள்ளதை, நகர்த்தப்பட்ட எண்ணுடன் கூட்ட வேண்டும். இது படம் (4.5b)இல் தெளிவாக காட்டப்பட்டுள்ளது. உதாரணத்திற்கு 5000H அடுக்குப் பகுதி பதிவேட்டில் இருப்பதாகக் கொள்வோம். இந்த எண்ணை இடது பக்கமாக நான்கு பிட் நகர்த்தினால், 50000H என்ற எண் கிடைக்கும். இப்பொழுது FFE0H என்ற முகவரி, அடுக்குச் சுட்டிக்காட்டிப் பதிவேட்டில் இருப்பதாகக் கொள்வோம். இதை 50000H உடன் கூட்டினால் 5FFE0H என்ற 20பிட் முகவரி கிடைக்கும். இது அடுக்கு உச்சியின் (Stack Top) தரவு சார்ந்த முகவரியாகும். (Physical address) இந்தத் தரவு சார்ந்த முகவரியைத் தனி எண்ணாகவோ அல்லது SS:SP அதாவது 5000:FFE0H என்றோ குறிப்பிடலாம்.



படம் (4.5) SS மற்றும் SP ஆகியவற்றைக் கூட்டி, அடுக்கு உச்சியின் தரவு சார்ந்த முகவரியைக் கணக்கிடுதல்

இதர சுட்டிக்காட்டி மற்றும் பதிவேடுகள்

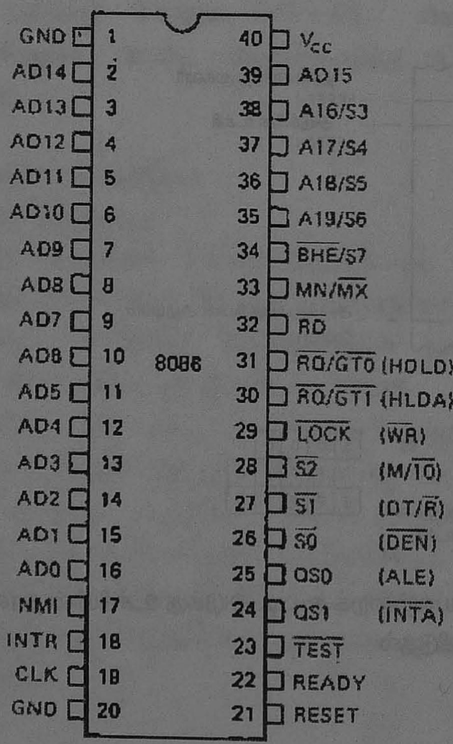
(Other pointer and register)

அடுக்குச் சுட்டிக்காட்டிப் பதிவேடு தவிர, செயற்பாட்டு அலகில் 16-பிட் அடித்தள சுட்டிக்காட்டிப் பதிவேடு (Base Pointer (BP) Register), 16-பிட் மூலச் சுட்டிக்காட்டிப் பதிவேடு (Source Index (SI) Register) மற்றும் 16பிட் சேர்ப்பிடச் சுட்டிக்காட்டிப் பதிவேடு (Destination Index (DI) Register) ஆகியவை உள்ளன. இந்த மூன்று பதிவேடுகளும், தரவுகளைத் தற்காலிகமாகச் சேமித்து வைக்கப் பயன்படுகின்றன.

4.3 இன்டெல் 8086 நுண்முறைப்படுத்தியின் முனை அமைப்பு

(Pin configuration of Intel 8086 Microprocessor)

இன்டெல் 8086இன் மின்முனை அமைப்பு, படம் (4.6)இல் காட்டப்பட்டுள்ளது. இது 40 முனைகள் கொண்ட ஒரு தொகுப்புச் சுற்றாகும். இது வேலை செய்வதற்கு, +5V நேரடி மின்னழுத்தம் தேவைப்படுகிறது. ஒவ்வொரு முனையிலிருந்து வெறியேறுகிற மற்றும் முனையின் வழியாக மையச் செயலகத்தினுள் செல்லுகிற சைகைகளைப் பற்றிய விளக்கத்தைக் காண்போம்.



படம் (4.6) 8086இன் மின்முனை அமைப்பு

$AD_0 - AD_{15}$: (முனை 2 முதல் 16 வரை மற்றும் முனை 39) இந்த முனைகள், முகவரி / தரவு ஊர்தியாகச் செயல்படுகின்றன. 8086 நுண்முறைப்படுத்தியில் நினைவக இருப்பிடத்தின் முகவரி 20-பிட்டுகளாகும். இதில் முதல் 16-பிட்டுகள் (தாழ்நிலை பிட்டுகள்), கருவிச் சுற்றின் தொடக்கத்தில், $AD_0 - AD_{15}$ கம்பிகளின் வழியாக மையச் செயலகத்தினால் அனுப்பப்படும். கருவிச் சுற்றின் பிற்பகுதியில், இந்தக் கம்பிகளின் வழியாகத் தரவுகள் அனுப்பப்படும். இந்த நுணுக்கத்தைப் பலவின் ஒன்றாக்குதல் (Multiplexing) எனலாம்.

ALE (Address Latch Enable) : (முனை 25)

இது முகவரி தாழ் செயலியக்கச் சைகையாகும். இந்தச் சைகை 8086 ஆல் வெளிச் சுற்றுகளுக்கு அனுப்பப்படும். இந்தச் சைகையின் மூலம், AD ஊர்தியில் உள்ளவை நினைவக முகவரி என்று தெரிந்து கொள்ளலாம்.

A16/S3 – A19/S6 : (முனை 35 முதல் 38 வரை)

இந்தக் கம்பிகளின் வழியாக, நினைவக இருப்பிட முகவரியின் கடைசி நான்கு பிட்டுகள் (உயர்நிலை பிட்டுகள்) அனுப்பப்படும். இவை கருவிச் சுற்றின் முதல் சுற்றில் அனுப்பப்படும். கருவிச் சுற்றின் பிற்பகுதியில், இதே கம்பிகளின் வழியாக, எந்த வேலை செய்ய வேண்டும் என்று உணர்த்தக் கூடிய நிலைப்பாடு சைகைகள் எடுத்துச் செல்லப்படும்.

$\overline{RD}$: (முனை 32)

இந்தச் சைகை, படிக்கும் செயலைக் கட்டுப்படுத்துகிறது. 8086 நுண்முறைப்படுத்தி, நினைவகத்திலிருந்தோ அல்லது உள்ளீடு / வெளியீடு நிலையத்திலிருந்தோ தரவுகளைப் படிக்கும்போது, இந்தச் சைகை தாழ்நிலையில் இருக்கும்.

$\overline{WR}$: (முனை 29)

இந்தச் சைகை தாழ்நிலையில் இருக்கும்போது, நுண்முறைப்படுத்தியானது நினைவகம் மற்றும் உள்ளீடு / வெளியீடு நிலையங்களில் எழுதுதல் செயலைச் செய்யும். அதிகபட்ச முறையில் (Maximum Mode), இந்த முனையின் மூலம் $\overline{LOCK}$ என்ற சைகை அனுப்பப்படும்.

M/$\overline{IO}$: (முனை 28)

இந்தச் சைகையின் மூலம், நினைவகத் தொடர்பான செயல்கள் நடைபெறுகின்றனவோ அல்லது உள்ளீடு / வெளியீடு தொடர்பான செயல்கள் நடைபெறுகின்றனவோ என்பதைப் புறச்சுற்றுகளுக்குத் தெரியப்படுத்தலாம். இந்தச் சைகை உயர்நிலையில் இருந்தால், நினைவகம் தொடர்பான செயல்கள்

நடைபெறுகின்றன என்றும், தாழ்நிலையில் இருந்தால் உள்ளீடு / வெளியீடு தொடர்பான செயல்கள் நடைபெறுகின்றன என்றும் உணரலாம்.

$\overline{BHE}$/S7 (Bus High Enable / Status) : (முனை 34)

இது ஊர்தி உயர் செயலியக்கச் சைகையாகும். தரவு ஊர்தியின் பெரும முக்கியத்துவம் வாய்ந்த பகுதியான $D_8 - D_{15}$ வழியாகத் தரவுப் பரிமாற்றம் நடைபெற, இந்தச் சைகை பயன்படுத்தப்படுகிறது. இதே முனையின் வழியாக S7 என்ற நிலைப்பாட்டுச் சைகையும் அனுப்பப்படுகிறது. இது உயர்நிலைச் சைகையாகும்.

DT/$\overline{R}$ (Data transfer /Receive) : (முனை 27)

இது தரவு அனுப்புதல் / பெறுதல் சைகையாகும். இந்தச் சைகை உயர்நிலையில் உள்ளபோது, ஊர்தியானது தரவை அனுப்பும் நிலையில் இருக்கும். தாழ்நிலையில் இருக்கும்போது, ஊர்தியானது தரவைப் பெறும் நிலையில் இருக்கும். அதிகபட்ச முறையில் (Maximum Mode), இந்த முனை வழியாக S1 என்ற நிலைப்பாட்டுச் சைகை அனுப்பப்படும்.

$\overline{DEN}$ (Data Enable Signal) : (முனை 26)

இது தரவு செயலியக்கச் சைகையாகும். தரவு ஊர்தி மூலம் தரவுப் பரிமாற்றம் நடைபெறும்போது, இந்தச் சைகையைப் பயன்படுத்தித் தகுதியான தரவைத் தரவு ஊர்தியில் வைக்கலாம். அதிகபட்ச முறையில் SO என்ற நிலைப்பாட்டுச் சைகை, இந்த முனை வழியாக அனுப்பப்படும்.

HOLD மற்றும் HLDA : (முனை 31 மற்றும் 30)

8086 நுண்முறைப்படுத்தியின் குறைந்தபட்ச முறை அமைப்பின் (Minimum Mode System) நேரடியாக நினைவகத்தை அணுகும் இடைப் பரிமாற்று சுற்றிற்கு HOLD மற்றும் HLDA சைகைகள் தேவைப்படுகின்றன. இதே முனைகள், அதிகபட்ச முறையில் (Maximum Mode) $\overline{RQ}/\overline{GT0}$ மற்றும் $\overline{RQ}/\overline{GT1}$ (Request / Grant Bus Access Control) சைகைகளாகவும் செயல்படுகின்றன.

INTR, $\overline{INTA}$ மற்றும் NMI : (முனை 18, 24 மற்றும் 17)

INTR : தடங்கல் வேண்டுதல் சைகை

$\overline{INTA}$: தடங்கல் சைகையைப் பெற்றதற்கு அடையாளமாகத் திரும்ப அனுப்பப்படும் சைகை

NMI : மறைக்க முடியாத தடங்கல் சைகை

இந்தச் சைகைகள், 8086 நுண்முறைப்படுத்திச் செயல்பாட்டில் தடங்கல் ஏற்படுத்துவதற்காகப் பயன்படுகின்றன. அதிகபட்ச முறையில் (Maximum Mode) முனை 24, QSI (வரிசை நிலைபாடு) சைகையாகச் செயல்படுகிறது.

TEST : (முனை 23)

இந்தச் சைகை உயர்ந்த நிலையில் இருக்கும்போது, நுண்முறைப்படுத்தி தன்னுடைய செயற்பாட்டைத் தற்காலிகமாக நிறுத்திவிடும். இந்தச் சைகை தாழ்நிலைக்குச் செல்லும்வரை, கட்டளைகளைச் செயல்படுத்துவதையும் நிறுத்திவிடும்.

READY : (முனை 22)

நினைவகம் அணுகும் செயற்பாட்டின்போது, தரவுகளோ அல்லது இயக்கக் குறியீடுகளோ நினைவகத்திலிருந்து வரும்வரை காத்திருக்கக்கூடிய நிலையை, இந்தச் சைகையின் மூலம் அடையலாம்.

RESET : (முனை 21)

இந்தச் சைகை உயர்நிலையில் இருந்தால், 8086 நுண்முறைப்படுத்தி அமைவு பெறும். அமைவு நிலையில், முதல் கட்டளை $FFFF0_H$ என்ற முகவரியையுடைய நினைவக இருப்பிடத்திலிருந்து கொணரப்படும்.

CLOCK : (முனை 19)

8086 செயற்பாட்டிற்குக் கடிகாரத் துடிப்புகள் தேவைப்படும். அதிக அதிர்வெண் உடைய கடிகாரத் துடிப்புகள் (5MHZ / 8MHZ/ 10MHZ) முனை 19இன் வழியாகச் செலுத்தப்படும்.

VCC : முனை 40இன் வழியாக +5v நேர்மின்னழுத்தம் மைய செயலகத்திற்கு கொடுக்கப்படுகிறது.

GND : முனை 1 மற்றும் 20, தரை இணைப்பு செய்யப்படுகிறது.

4.4 8086 நுண்முறைப்படுத்தி செயல்படும் முறைகள் (Operating modes of 8086 Microprocessor)

இன்டெல் 8086 நுண்முறைப்படுத்தியின் செயல்படும் முறைகள், இரண்டு வகைப்படும்.

1. குறைந்த பட்ச முறை (Minimum Mode)
2. அதிக பட்ச முறை (Maximum Mode)

நுண்கணினி அமைப்பில், ஒரே ஒரு 8086 மையச்செயலகத்தைப் பயன்படுத்தினால், அவற்றின் செயல்பாடு குறைந்தபட்ச முறையில் இருக்கும். இந்த முறையில் மையச் செயலகமானது, நினைவகம் மற்றும் உள்ளீடு / வெளியீடு

சாதனங்களுக்குத் தேவையான கட்டுப்பாடு சைகைகளை வெளியிடும். பலமுறைப்படுத்தி அமைப்பில், 8086 நுண்முறைப்படுத்தியின் செயல்பாடு அதிகபட்ச முறையில் இருக்கும். இந்த முறையில், கட்டுப்பாடு சைகைகள், இன்டெல் 8288 ஊர்தி கட்டுப்பாட்டகத்தால் வெளியிடப்படும். இன்டெல் 8086 நுண்முறைப்படுத்தியில் கட்டுப்பாட்டுச் சைகைகளை உற்பத்தி செய்வதற்காகவே இன்டெல் 8288 ஊர்திக் கட்டுப்பாட்டகம் பயன்படுத்தப்படுகிறது.முனை $MN/\overline{MX}$, 8086இன் முறைகளை முடிவு செய்கிறது. இந்த முனை உயர்நிலையில் இருந்தால், மையச் செயலகமானது குறைந்தபட்ச முறையில் பயன்படுத்தப்படும். இதே முனை தாழ்நிலையில் இருந்தால் மையச் செயலகமானது அதிகபட்ச முறையில் பயன்படுத்தப்படும். முனை 24 முதல் முனை 31 வரை இரண்டு விதமான சைகைகள் வெளியிடப்படுகின்றன. அதில் ஒரு தொகுப்பு, 8086 மையச்செயலகம் குறைந்தபட்ச முறையில் செயல்படும்போது பயன்படுத்தப்படுகிறது. மற்றொரு தொகுப்பு, அதிகபட்ச முறையில் செயல்படும்போது பயன்படுத்தப்படுகிறது.

4.4.1 குறைந்தபட்ச முறைக்கான மின்முனை அமைப்பு (Pin Description for Minimum Mode)

இந்த முறையில் முனை $MN / \overline{MX}$ + 5V நேர் மின்னழுத்தத்துடன் இணைக்கப்படும். 24 முதல் 31 வரையிலான முனைகளைப் பற்றிய விளக்கம், கீழே கொடுக்கப்பட்டுள்ளது.

$\overline{INTA}$ (வெளியீடு) (முனை 24)

தடங்கல் சைகையைப் பெற்றுக் கொண்டதற்கு அடையாளமாக, இன்டெல் 8086, நுண்முறைப்படுத்தியால் திரும்ப அனுப்பப்படும் சைகையாகும். இதன் செயல்பாடு தாழ் நிலையில் இருக்கும்.

ALE (வெளியீடு) (முனை 25)

இது முகவரி தாழ் செயலியக்கச் சைகையாகும். கருவிச் சுற்றின் முதல் கடிகாரச் சுற்றின்போது, இந்தச் சைகை உயர்நிலைக்கு செல்லும். இதனால் இன்டெல் 8282/8283 தாழில் (Latch) முகவரி வைக்கப்படும்.

$\overline{DEN}$ (வெளியீடு) (முனை 26)

இது தரவுச் செயலியக்கச் சைகையாகும். இதன் இயக்கம் தாழ் நிலையில் இருக்கும்.

$DT/\overline{R}$ (வெளியீடு) (முனை 27)

இது தரவு அனுப்புதல் / பெறுதல் சைகையாகும். இது உயர்நிலையில் இருந்தால், தரவுகள் மையச்செயலகத்திலிருந்து வெளியே அனுப்பப்படும். இது தாழ் நிலையில் இருந்தால், தரவுகள் பெறப்படும்.

$M/\overline{IO}$ (வெளியீடு) (முனை 28)

நினைவகம் மற்றும் உள்ளீடு / வெளியீடு சாதனங்களை அணுகப் பயன்படுத்தப்படும் சைகை. இந்தச் சைகை உயர்நிலையில் இருந்தால், மையச் செயலகம் நினைவகத்தை அணுக விரும்புகிறது என்று பொருள். தாழ் நிலையில் இருந்தால், மையச் செயலகம் உள்ளீடு / வெளியீடு சாதனங்களை அணுக விரும்புகிறது என்று பொருள்.

$\overline{WR}$ (வெளியீடு) (முனை 29)

இந்தச் சைகை தாழ்நிலையில் இருக்கும்போது, மையச் செயலகம், நினைவகம் மற்றும் உள்ளீடு / வெளியீடு சாதனங்களில் எழுதுதல் செயலைச் செய்யும்.

HOLD (உள்ளீடு) (முனை 31)

நுண்கணினி அமைப்பில் மையச்செயலகத்தைத் தவிரப் பிற சாதனங்கள் முகவரி மற்றும் தரவு ஊர்தியைப் பயன்படுத்த விரும்பினால், அவை முனை 31இன் வழியாக HOLD வேண்டுதல் சைகையை மையச்செயலகத்திற்கு அனுப்பும். இந்தச் சைகை உயர்நிலையில் செயல்படும்.

HLDA (வெளியீடு) (முனை 30)

HOLD, சைகையைப் பெற்றுக் கொண்டதற்கு அடையாளமாக நுண்முறைப் படுத்தி, திரும்ப அனுப்பும் சைகையாகும். இந்தச் சைகையும் உயர்நிலையில் செயல்படும்.

4.4.2 அதிகபட்ச முறைக்கான மின்முனை அமைப்பு (Pin description for maximum mode)

அதிகபட்ச முறை செயற்பாட்டிற்கு, முனை $MN/\overline{MX}$ தாழ் நிலையில் இருக்கும். இந்த முனை தரையிறக்கம் செய்யப்படும். 24 முதல் 31 வரையிலான முனைகளைப் பற்றிய விளக்கம், கீழே கொடுக்கப்பட்டுள்ளது.

QS_1, QS_0 (வெளியீடு) (முனை 24, 25)

இவை கட்டளை வரிசை (Instruction Queue) நிலைப்பாட்டுச் சைகைகள். இந்தச் சைகைகளின் வாதியில் நிலை, கீழே கொடுக்கப்பட்டுள்ளது.

QS_1	QS_0	
0	0	எந்தச் செயற்பாடும் நடக்காது.
0	1	வரிசையிலிருந்து இயக்கக் குறியீட்டின் முதல் பைட்டைக் கொணர்தல்
1	0	கட்டளைப் பரிமாற்றம் நடைபெறுவதால், வரிசையைக் காலியாக வைத்தல்
1	1	வரிசையிலிருந்து இயக்கக் குறியீட்டின் மற்ற பைட்டுகளைக் கொணர்தல்

$\overline{S}_0, \overline{S}_1, \overline{S}_2$ (வெளியீடு) (முனை 26,27,28)

இவை நிலைப்பாட்டுச் சைகைகளாகும். இந்தச் சைகைகள், இன்டெல் 8288 ஊர்திக் கட்டுப்பாட்டகத்துடன் இணைக்கப்படும். ஊர்திக் கட்டுப்பாட்டகம், நினைவகம் மற்றும் உள்ளீடு / வெளியீடு சாதனங்களுக்குத் தேவையான கட்டுப்பாட்டுச் சைகைகளை உற்பத்தி செய்யும். $\overline{S}_0, \overline{S}_1$ மற்றும் $\overline{S}_2$ சைகைகளின் வாதியியல் நிலை கீழே கொடுக்கப்பட்டுள்ளது.

$\overline{S}_2$	$\overline{S}_1$	$\overline{S}_0$	
0	0	0	தடங்கல் சைகையை ஒப்புக் கொள்ளுதல்
0	0	1	உள்ளீடு / வெளியீடு நிலையத்திலிருந்து தரவைப் படித்தல்
0	1	0	உள்ளீடு / வெளியீடு நிலையத்திலிருந்து தரவை எழுதுதல்
0	1	1	செயற்பாட்டை நிறுத்துதல்
1	0	0	இயக்கக் குறியீட்டைக் கொணர்தல்
1	0	1	நினைவகத்தைப் படித்தல்
1	1	0	நினைவகத்தில் எழுதுதல்
1	1	1	எந்த செயல்பாடும் இல்லாத நிலை (Passive State)

$\overline{\text{LOCK}}$ (வெளியீடு) (முனை 29)

இது தாழ்நிலைச் செயற்பாட்டுச் சைகையாகும். இது தாழ்நிலையில் இருக்கும் போது, எல்லாத் தடங்கல் சைகைகளும் மறைக்கப்படும். எந்த ஒரு வேண்டுதல் சைகையும் ஏற்றுக் கொள்ளப்படமாட்டாது.

$\overline{\text{RQ}}/\overline{\text{GT}}_1, \overline{\text{RQ}}/\overline{\text{GT}}_0$ (முனை 30, 31)

மையச் செயலகத்தைப் பிறமுறைப்படுத்திகள், ஊர்திகளை வெளியிடச் சொல்லி, இந்தச் சைகைகளை முனை 30 மற்றும் 31இன் வழியாக அனுப்பும். இந்த சைகைகளில் $\overline{\text{RQ}}/\overline{\text{GT}}_0$ சைகைகளுக்கு அதிக முக்கியத்துவம் கொடுக்கப்படுகிறது.

அதிகபட்ச முறையில் $\overline{WR}$, ALE, $\overline{DEN}$ மற்றும் DT/$\overline{R}$ சைகைகள், முறைப்படுத்தியிலிருந்து நேரடியாகக் கிடைப்பதில்லை. இவை கட்டுப்பாட்டகம் 8288 லிருந்து கிடைக்கின்றன.

4.5 இன்டெல் 8086 நுண்முறைப்படுத்தியின் கட்டளை வகைகள்

4.5.1 தரவு மாற்றுக் கட்டளைகள் (Data Transfer Instructions)

இந்த வகைக் கட்டளைகள், பதிவேடுகளுக்கிடையே அல்லது பதிவேடு மற்றும் நினைவகத்திற்கிடையே தரவுகளை மாற்றம் செய்யும்.

(உ.ம்) MOV, XCHG, XLAT, LEA, LDS மற்றும் LES கட்டளைகள்.

MOV கட்டளைகள்

இந்த வகைக் கட்டளைகள், தரவு அல்லது இயமியை, மூலத்திலிருந்து (Source) சேர்ப்பிடத்திற்கு (Destination) மாற்றம் செய்ய உதவுகின்றன. இயமிகள் உள்ளார்ந்த பதிவேடுகளிலோ அல்லது நினைவக இருப்பிடத்திலோ இருக்கும். உடனடி முகவரியிடல் வகைகளில் (Immediate Addressing Modes), ஒரு இயமி மூலமாக செயல்படும் சில MOV கட்டளைகளைப் பற்றிக் கீழே காண்போம்.

MOV AL, BL → BLஇன் உள்ளடக்கம் ALக்கு நகர்த்த வேண்டும்.

MOV BX, CX → CXஇன் உள்ளடக்கம் BX க்கு நகர்த்த வேண்டும்.

MOV BL, 36H → 36H என்ற தரவை BL க்கு நகர்த்த வேண்டும்.

MOV CX, 1234H → 1234H என்ற தரவை CX க்கு நகர்த்த வேண்டும்.

MOV BL, [1200H] → தரவுப் பகுதியில் (Data Segment) 1200H என்ற இடப்பெயர்ச்சி முகவரியிலுள்ள (Offset or Displacement address) பைட்டை BL க்கு நகர்த்த வேண்டும்.

MOV DL, [SI] → இந்தக் கட்டளையைப் பின்வருமாறு குறிப்பிடலாம். DL←(DS:SI). அதாவது DS:SI இன் உள்ளடக்கம் DL க்கு நகர்த்த வேண்டும்.

MOV CS, AX → AXலிருந்து ஒரு வார்த்தையை CS க்கு நகர்த்த வேண்டும்.

MOV CS, 1000H → ஒரு பைட்டைப் பகுதிப் பதிவேட்டிற்கு (Segment Register) நகர்த்த முடியாது.

XCHG கட்டளைகள்

பரிமாற்றுக் கட்டளைகள், இரண்டு பதிவேடுகளின் உள்ளடக்கத்தைப் பரிமாற்றம் செய்யும் அல்லது நினைவகங்கள் மற்றும் பதிவேடுகளுக்கிடையே தரவுகளைப்

பரிமாற்றம் செய்யும். இந்த வகை கட்டளைகளில், பகுதிப் பதிவேடுகளைப் பயன்படுத்தமுடியாது.

(உ.ம்) XCHG AX, CX : (AX) ↔ (CX)
XCHG BL, AL : (BL) ↔ (AL)
XCHG AL, [1200H] : (AL) ↔ (DS : 1200H)

XLAT கட்டளை

மொழி பெயர்ப்புக் கட்டளை (Translate Instruction), பார்வை அட்டவணை (Look-up table) முறையில் பயன்படுத்தப்படுகிறது. இந்தக் கட்டளைகளில் BX மற்றும் AL பதிவேடுகள் பயன்படுத்தப்படுகின்றன. இந்த கட்டளை, பின்வருமாறு செயல்படுகிறது. BX = 1000H மற்றும் AL = 05H எனக் கருதுவோம். (DS : 1005H)இல் 19_H என்ற தரவு இருப்பதாகக் கருதுவோம். XLAT கட்டளையைச் செயல்படுத்தும்போது, DS : BX + AL என்ற தரவு சார்ந்த முகவரியையுடைய (Physical Address) நினைவகத்தில் உள்ள தரவு, AL பதிவேட்டிற்கு மாற்றப்படும். அதாவது 19_H AL பதிவேட்டிற்கு மாற்றப்படும்.

PA = (DS)0 + (BX) + (AL)

PA = (DS)0 + 1000_H + 05

(AL) ← (DS: BX + AL)

LEA கட்டளை

இந்தக் கட்டளைப்படி, செயலூக்கம் உடைய முகவரியை (Effective address or offset) கட்டளையில் குறிப்பிடப்பட்டுள்ள 16-பிட் பதிவேட்டில் பளுவேற்ற வேண்டும். (உ.ம்) LEA BX, [SI] இந்தக் கட்டளைப்படி, SI இல் உள்ள EAஐ, BX பதிவேட்டில் பளுவேற்ற வேண்டும்.

LDS கட்டளை

இந்தக் கட்டளைப்படி, நினைவகத்திலிருநது எடுக்கப்பட்ட வார்த்தைகளை (Words), ஒரு பதிவேட்டிலோ அல்லது தரவுப் பகுதியிலோ பளுவேற்ற வேண்டும். (உ.ம்) LDS SI, [BX] அடுத்தடுத்துள்ள நான்கு நினைவக இருப்பிடங்களின் உள்ளடக்கத்தை, SI பதிவேடு மற்றும் தரவுப் பகுதியில் சேமிக்க வேண்டும். செயலூக்கமுடைய முகவரி (Effective address), BX பதிவேட்டில் இருக்கும்.

DS = 0200_H, SI = 1350_H, BX = 2300_H

PA = (DS)0 + (BX)

= 02000_H + 2300_H

= 04300_H

தரவுகள் பின்வரும் நினைவக இருப்பிடங்களில் சேமிக்கப்பட்டிருக்கும்.

04300	00
04301	15
04302	00
04303	03

மேற்குறிப்பிடப்பட்ட கட்டளையைச் செயல்படுத்திய பிறகு, SI = 1500_H மற்றும் DS = 0300_H ஆகும்.

LES கட்டளை

இந்தக் கட்டளைப்படி ஒரு பதிவேடு மற்றும் கூடுதல் பகுதியைப்(ES) பளுவேற்ற வேண்டும். (உ.ம்) LES DI, [BX]

4.5.2 கணிதக் கட்டளைகள் (Arithmetic Instructions)

8086 நுண்முறைப்படுத்தியில் கூட்டல், கழித்தல், பெருக்கல் மற்றும் வகுத்தல் போன்ற செயல்களுக்குத் தனித்தனிக் கட்டளைகள் உள்ளன.

ADD கட்டளை

பொதுவாகக் கூட்டல் கட்டளைகள், கீழ்க்கண்ட அமைப்பில் இருக்கும்.

ADD சேருமிடம், மூலம்

இந்தக் கட்டளை அமைப்பின் மூலம் (Source) என்பது, ஒரு எண்ணாகவோ, ஒரு பதிவேடாகவோ அல்லது நினைவக இருப்பிடமாகவோ இருக்கலாம். அதேபோன்று சேருமிடம் (Destination), ஒரு பதிவேடாகவோ அல்லது நினைவக இருப்பிடமாவோ இருக்கலாம். மூலம் மற்றும் சேருமிடத்தின் உள்ளடக்கம் கூட்டப்பட்டு அதன் முடிவு சேருமிடத்தில் வைக்கப்படும். இயமியானது 8-பிட் அல்லது 16-பிட் ஆக இருக்கலாம். சேருமிடம் மற்றும் மூலம் இரண்டுமே, நினைவக இருப்பிடமாக இருக்க முடியாது.

(உ.ம்) ADD AL, BL
ADD BX, CX
ADD AL, 75H
ADD DX, [SI]

ADC கட்டளை

இந்தக் கட்டளைப்படி, சேருமிடம் மற்றும் மூலத்தின் உள்ளடக்கத்துடன் சுமைக்கொடியின் உள்ளடக்கமும் சேர்ந்து கூட்டப்பட்டு, முடிவு சேருமிடத்தில் வைக்கப்படும். (உ.ம்) ADC AL, BL:(AL)← (AL) + (BL) + CF

INC கட்டளை

இந்தக் கட்டளைப்படி, கட்டளையில் குறிப்பிட்டுள்ள பதிவேட்டின் உள்ளடக்கம் ஒரு பிட் அதிகரிக்கப்படும். (உ.ம்) INC BL, INC BX

DAA கட்டளை

(DAA – Decimal Adjust Accumulator)

இரண்டு ஈரடிக் குறியீடுகளுடன் கூடிய பத்தடி எண்களைக் (BCD Numbers) கூட்டும்போது, முடிவை BCD எண்ணாக மாற்ற, இந்தக் கட்டளை பயன்படுத்தப்படுகிறது. இந்தக் கட்டளை AL பதிவேட்டில் உள்ள முடிவின்மீது செயல்படுத்தப்படும். முடிவின் சிறும முக்கியத்துவம் வாய்ந்த பிட்டுகள் 9ஐ விட பெரிதாக இருந்தால் 06 என்ற எண் ALஇல் உள்ள முடிவுடன் கூட்டப்படும். பெரும முக்கியத்துவம் வாய்ந்த பிட்டுகள் 9ஐ விட பெரிதாக இருந்தால், 60_H என்ற எண் ALஇல் உள்ள முடிவுடன் கூட்டப்படும்.

AAA கட்டளை

(AAA – ASCII Adjust for Addition)

(ASCII - அமெரிக்கச் செய்திப் பரிமாற்றப் படித்தரக் குறியீடு)

பத்தடி எண்கள் (0 முதல் 9 வரை), ASCII முறையில் 30_H முதல் 39_H வரை குறிப்பிடப்படும். ASCII முறையில் குறிப்பிடப்பட்ட இரண்டு பத்தடி எண்களை, 8086 நுண்முறைப்படுத்தி மூலம் கூட்டலாம். முடிவு ஈரடிக் குறியீடுகளுடன் கூடிய பத்தடி எண்களாக (BCD), AAA கட்டளை கொண்டு மாற்றப்படுகிறது. இந்தக் கட்டளை, கூட்டல் செயலுக்குப் பிறகு பயன்படுத்தப்படுகிறது. இந்தக் கட்டளையில் சேருமிடம் AL பதிவேடாக மட்டும்தான் இருக்க முடியும்.

கழித்தல் கட்டளைகள்

SUB கட்டளை

இந்த கட்டளைப்படி, மூலத்தின் இயமியைச் சேருமிடத்திலுள்ள இயமியிலிருந்து கழித்து முடிவைச் சேருமிடத்தில் வைக்க வேண்டும்.

(உ.ம்) SUB AL, BL : (AL) ← (AL) – (BL)
SUB BX, 1234H : (BX) ← (BX) – 1234H
SUB CX, [1234H] : (CX) ← (CX) – (DS : 1234H)

SBB கட்டளை

இந்தக் கட்டளைப்படி, மூலத்திலுள்ள இயமியைச் சுமையுடன் சேர்த்துச் சேருமிடத்திலுள்ள இயமியுடன் கழித்து, முடிவைச் சேருமிடத்தில் வைக்க வேண்டும்.

(உ.ம்) SBB AX, BX : (AX) ← (AX) – (BX) – CF

SBB AX, [SI] : (AX) ← (AX) – (DS:SI) – CF

DEC கட்டளை

கட்டளையில் குறிப்பிடப்பட்டுள்ள பதிவேட்டில் உள்ளதை, ஒரு பிட் குறைக்க வேண்டும். (உ.ம்) DEC CL : (CL) ← (CL) – 1

DAS கட்டளை

(DAS – Decimal Adjust for Subtraction)

BCD கூட்டலில் DAA கட்டளை எவ்வாறு பயன்படுத்தப்பட்டதோ, அதே போன்று, BCD கழித்தலில் DAS கட்டளை பயன்படுத்தப்படுகிறது.

AAS கட்டளை

(AAS – ASCII Adjust for Subtraction)

இந்தக் கட்டளையின் மூலம், ASCII முறையில், குறிப்பிடப்பட்ட இரண்டு பத்தடி எண்களைக் கழிக்கலாம்.

NEG கட்டளை

எதிர்மறைக் கட்டளையானது, ஒரு நேர்க்குறியீடு எண்ணை எதிர்க் குறியீடு எண்ணாக மாற்றும். அதாவது, ஒரு எண்ணை இரண்டின் நிரப்பியாக (2'S Complement) மாற்றும்.

(உ.ம்) NEG BL

BL பதிவேட்டில் $2A_H$ என்ற தரவு இருப்பதாகக் கருதுவோம்.

அதாவது BL = 0010 1010

BLஇன் ஒன்றின் நிரப்பு = 1101 0101

BLஇன் இரண்டின் நிரப்பு = 1101 0110

NEG BL என்ற கட்டளை, BL பதிவேட்டில் உள்ள $2A_H$ என்ற தரவை, இரண்டின் நிரப்பு முறையில் $D6_H$ ஆக மாற்றம் செய்கிறது.

பெருக்கல் கட்டளைகள்

MUL கட்டளை

இந்தக் கட்டளை, இரண்டு எண்களைப் பெருக்கப் பயன்படுகிறது. 8-பிட் பெருக்கலுக்கு, இயமிகளில் ஒரு இயமி, AL பதிவேட்டில் இருக்கும். மற்றொரு இயமி, 8-பிட் பதிவேட்டிலோ அல்லது நினைவக இருப்பிடத்திலோ இருக்கும். இரண்டு 8-பிட் எண்களை பெருக்கினால்,முடிவு 16-பிட்டாக இருக்கும். முடிவை AX பதிவேட்டில் வைக்க வேண்டும். (உ.ம்) MUL BH : (AX) ← (AL) x (BX) 16-பிட் பெருக்கலுக்கு, இயமிகளில் ஒரு இயமி AX பதிவேட்டில் இருக்கும். மற்றொரு இயமி, ஒரு பதிவேட்டிலோ அல்லது நினைவக இருப்பிடத்திலோ இருக்கும். முடிவு DX – AX பதிவேட்டு ஜோடியில் இருக்கும். (உ.ம்) MUL BX : (DX – AX) ←(AX) x (BX)

IMUL கட்டளை

இந்தக் கட்டளை, முழு எண்களின் பெருக்கலுக்குப் பயன்படுத்தப்படுகிறது. இரு முழு எண்களைப் பெருக்கும் போது முடிவு நேர்க்குறியீடாக இருந்தால், பெருக்கல் தொகை ஈரடி அமைப்பில் இருக்கும். முடிவு எதிர்க் குறியீடாக இருந்தால், பெருக்கல் தொகை இரண்டின் நிரப்பியாக(2'S complement) இருக்கும்.

AAM கட்டளை

(AAM – ASCII Adjust for Multiplication)

இந்தக் கட்டளை, இரண்டு ஈரடிக் குறியீடுகளுடன் கூடிய பத்தடி எண்களைப் பெருக்கிய பிறகு பயன்படுத்தப்படும். முடிவு AX பதிவேட்டில் இருக்கும்.

வகுத்தல் கட்டளைகள்

DIV கட்டளை

குறியிடப்படாத ஒரு வார்த்தையை ஒரு பைட்டால் வகுக்கவோ, அல்லது குறியிடப்படாத ஒரு இரட்டை வார்த்தையை ஒரு வார்த்தையால் வகுக்கவோ, இந்தக் கட்டளை பயன்படுகிறது. ஒரு வார்த்தையை ஒரு பைட்டால் வகுக்கும்போது, வார்த்தையானது AX பதிவேட்டில் இருக்கும். இதில் வார்த்தை வகுக்கப்படும் (Dividend) எண்ணாகவும், பைட் என்பது வகுக்கும் எண்ணாகவும் (Divisor) இருக்கும். வகுத்தலுக்குப் பிறகு ஈவு (Quotient) AL பதிவேட்டிலும், மீதி (Remainder) AX பதிவேட்டிலும் இருக்கும்.

(உ.ம்) DIV BL : AXஇல் உள்ள வார்த்தையை BLஇல் உள்ள பைட்டால்
வகுத்து, ஈவை AL லிலும் மீதியை AH லிலும் வைக்க வேண்டும்.

IDIV கட்டளை

இந்தக் கட்டளை, குறியுடன் கூடிய வார்த்தையைக் குறியுடன் கூடிய பைட்டால் வகுக்கவோ அல்லது குறியுடன் கூடிய இரட்டை வார்த்தையைக் குறியுடன் கூடிய வார்த்தையால் வகுக்கவோ பயன்படுகிறது. இந்த வகுத்தலில், ஈவின் 8 பிட்டுகளில் ஒரு பிட், குறியீடுக்காக ஒதுக்கப்படுகிறது.

4.5.3 வாதியியல் கட்டளைகள்

(Logic Instruction)

AND,OR, EX-OR மற்றும் NOT வாதியியல் கட்டளைகள், இன்டெல் 8086இல் பயன்படுத்தப்படுகின்றன. இந்தக் கட்டளைகளில் மூல இயமி ஒரு பதிவேடாகவோ, நினைவகமாகவோ அல்லது தரவாகவோ இருக்கலாம். சேருமிட இயமி, ஒரு பதிவேடாகவோ அல்லது நினைவகமாகவோ இருக்கலாம்.

AND, OR, XOR கட்டளைகள்

(உ.ம்)

AND AL, BL : BL பதிவேட்டில் உள்ளவை AL பதிவேட்டில் உள்ளவற்றுடன் உம் (AND) வினைப்பட்டு வருகின்ற முடிவை, AL பதிவேட்டில் வைக்க வேண்டும்.

OR BL, CL : CL பதிவேட்டில் உள்ளவை BL பதிவேட்டில் உள்ளவற்றுடன் அல்ல (OR) வினைப்பட்டு வருகின்ற முடிவை, BL பதிவேட்டில் வைக்க வேண்டும்.

XOR AL, BL : BL பதிவேட்டில் உள்ளவை AL பதிவேட்டில் உள்ளவற்றுடன் தவிர்க்கப்பட்ட அல்ல வினைப்பட்டு வருகின்ற முடிவை, AL பதிவேட்டில் வைக்க வேண்டும்.

இந்த வாதியியல் கட்டளைகளில், XOR கட்டளை ஒரு பதிவேட்டைச் சுழியாக்குவதற்கு பயன்படுகிறது. (உ.ம்) XOR AX, AX. இந்தக் கட்டளைப்படி, AX பதிவேட்டின் உள்ளடக்கம் சுழியாக மாறும்.

NOT கட்டளை

எதிர்மச் செயற்பாடு, ஒரு இயமியைக் கொண்டு செயல்படும். இந்த இயமி, ஒரு உள்ளார்ந்த பதிவேட்டின் உள்ளடக்கமாகவோ அல்லது நினைவகத்தின் ஒரு இருப்பிடமாகவோ இருக்கலாம். (உ.ம்) NOT AL : (AL) ⟵ $(\overline{AL})$

TEST கட்டளை

இந்தக் கட்டளை, இரண்டு இயமிகளுக்கிடையே உம் வினையை ஏற்படுத்தும். இந்தக் கட்டளைக்கும் AND கட்டளைக்கும் உள்ள வேறுபாடு என்னவென்றால், AND கட்டளை சேருமிடத்தின் உள்ளடக்கத்தை மாற்றிவிடும். ஆனால் Test கட்டளை, சேருமிடத்தின் உள்ளடக்கத்தை மாற்றாது.

(உ.ம்)

TEST AL, BH : AL இன் உள்ளடக்கம் BH இன் உள்ளடக்கத்துடன் உ.ம் வினையை ஏற்படுத்தும். ஆனால் AL மற்றும் BH இன் உள்ளடக்கம் மாறாது.

TEST BL, 05 : BL இன் உள்ளடக்கம் 05 என்ற எண்ணுடன் உம் வினையை ஏற்படுத்தும். ஆனால் BL இன் உள்ளடக்கம் மாறாது.

4.5.4 திசை மாற்றுக் கட்டளைகள்

(SHIFT Instructions)

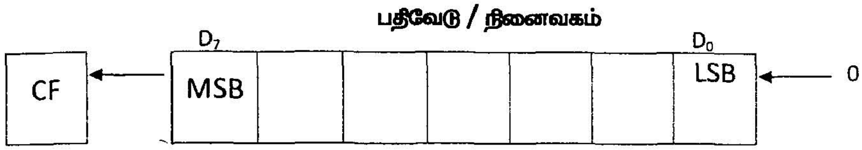
திசை மாற்றுக் கட்டளைகள் நான்கு வகைப்படும்.

1. SHL (Shift Logical Left)
2. SAL (Shift Arithmetic Left)
3. SHR (Shift Logical Right)
4. SAR (Shift Arithmetic Right)

SHL / SAL கட்டள்ளைகள்

இந்த இரண்டு கட்டளைகளுமே, ஒரே செயலைச் செய்கின்றன. இந்தக் கட்டளைகளின்படி, ஒரு பதிவேடு மற்றும் நினைவக இருப்பிடத்தின் உள்ளடக்கம், இடதுபக்கமாக நகர்த்தப்பட வேண்டும். இவ்வாறு நகர்த்தும்போது, சிறும முக்கியத்துவம் வாய்ந்த பகுதியில் சுழியைச் சேர்க்க வேண்டும். மற்றும் பெரும முக்கியத்துவம் வாய்ந்த பிட்டைச் சுமைக்கொடிக்கு(CF)நகர்த்த வேண்டும். இன்டெல்

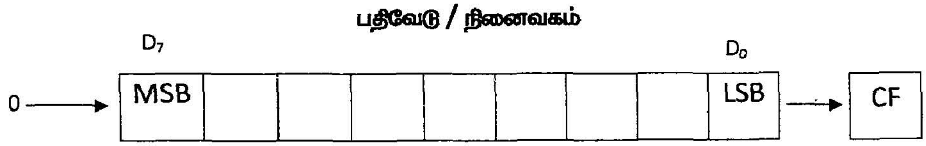
8086ஆல் ஒரு 6-பிட் பதிவேட்டின் உள்ளடக்கம் மற்றும் நினைவகத்திலுள்ள ஒரு வார்த்தையை, SHL / SAL கட்டளைப்படி இடதுபக்கமாக நகர்த்த முடியும்.



படம் 4.7

SHR கட்டளை

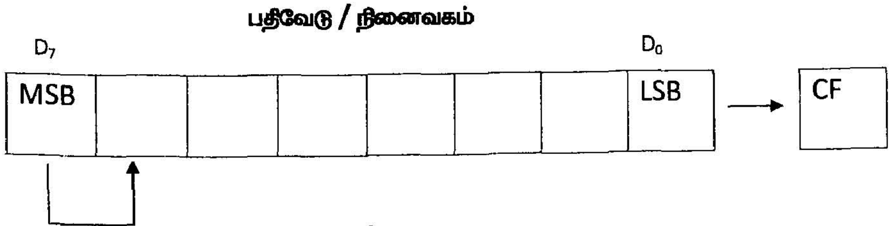
இந்தக் கட்டளைப்படி, ஒவ்வொரு பிட்டையும் வலதுபக்கமாக ஒரு பிட் நகர்த்த வேண்டும். சிறும முக்கியத்துவம் வாய்ந்த பிட்டைச் சுமைக்கொடிக்கு நகர்த்த வேண்டும். சுழியைப் பெரும முக்கியத்துவம் வாய்ந்த பிட்டில் சேர்க்க வேண்டும்.



படம் 4.8

SAR கட்டளை

இந்தக் கட்டளைப்படி ஒவ்வொரு பிட்டையும் வலது பக்கமாக நகர்த்த வேண்டும். சிறும முக்கியத்துவம் வாய்ந்த பிட், சுமைக் கொடிக்கு நகர்த்தப்படும். பெரும முக்கியத்துவம் வாய்ந்த பிட் வலதுபக்கமாக ஒரு பிட் நகர்த்தப்பட்டாலும், அந்த இடத்தை ஏற்கனவே இருந்த MSB பிட் நிரப்பும்.



படம் 4.9

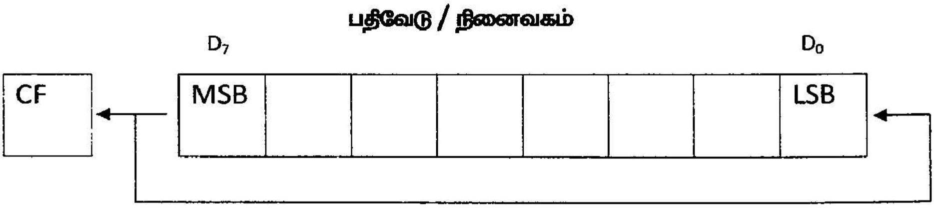
4.5.5 சுழற்சிக் கட்டளைகள்
(Rotate Instructions)

சுழற்சிக் கட்டளைகள் நான்கு வகைப்படும்.

1. ROL (Rotate Left)
2. RCL (Rotate Left Through Carry)
3. ROR (Rotate Right)
4. RCR (Rotate Right Through Carry)

ROL கட்டளை

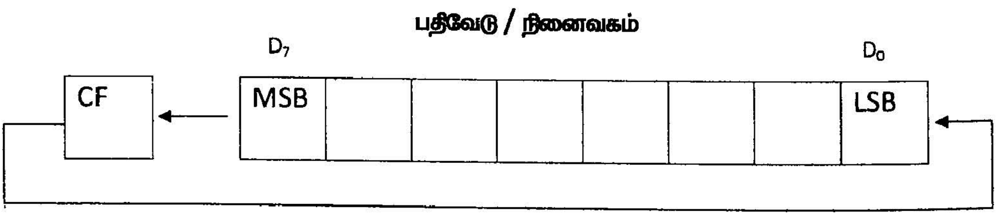
இந்தக் கட்டளைப்படி, ஒவ்வொரு பிட்டையும் இடதுபக்கமாக ஒரு பிட் நகர்த்த வேண்டும் (சுமையில்லாமல் நகர்த்த வேண்டும்). பெரும முக்கியத்துவம் வாய்ந்த பிட், சிறும முக்கித்துவம் வாய்ந்த பிட்டுக்கும் சுமைக் கொடிக்கும் நகர்த்தப்படும். இந்தக் கட்டளையைப் பயன்படுத்தி, ஒரு 16-பிட் பதிவேட்டின் உள்ளடக்கத்தையோ அல்லது நினைவகத்திலுள்ள வார்த்தையையோ நகர்த்தலாம்.



படம் 4.10

RCL கட்டளை

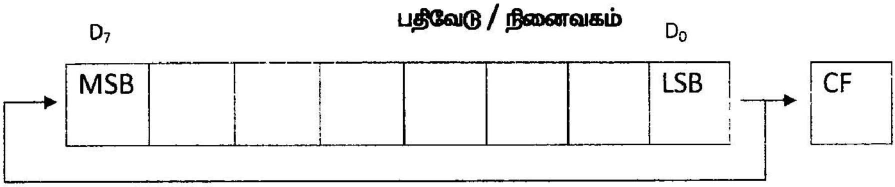
இந்தக் கட்டளைப்படி, ஒரு பதிவேடு மற்றும் நினைவகத்தின் உள்ளடக்கத்தை சுமையுடன் சேர்த்து ஒரு பிட் இடதுபக்கமாக நகர்த்த வேண்டும். அதாவது, பெரும முக்கியத்துவம் வாய்ந்த பிட் சுமைக் கொடிக்கும், சுமைக் கொடியில் உள்ள பிட் சிறும முக்கியத்துவம் வாய்ந்த பிட்டுக்கும் நகர்த்தப்பட வேண்டும்.



படம் 4.11

ROR கட்டளை

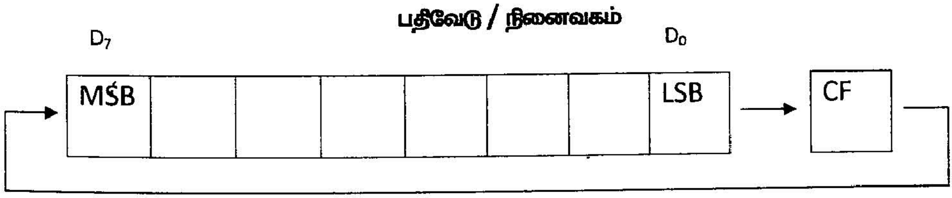
இந்தக் கட்டளைப்படி, ஒரு பதிவேடு மற்றும் நினைவகத்தின் உள்ளடக்கத்தை ஒரு பிட் வலதுபக்கமாக நகர்த்த வேண்டும். அதாவது சிறும முக்கியத்துவம் வாய்ந்த பிட்டைச் சுமைக் கொடிக்கும் மற்றும் பெரும முக்கியத்துவம் வாய்ந்த பிட்டுக்கு ம்நகர்த்த வேண்டும்.



படம் 4.12

RCR கட்டளை

இந்தக் கட்டளைப்படி, ஒரு பதிவேடு மற்றும் நினைவகத்தின் உள்ளடக்கத்தை சுமையுடன் சேர்த்து ஒரு பிட் வலதுபக்கமாக நகர்த்த வேண்டும். அதாவதுசி றும முக்கியத்துவம் வாய்ந்த பிட் சுமைக் கொடிக்கும், சுமைக் கொடியிலுள்ள பிட் பெரும முக்கியத்துவம் வாய்ந்த பிட்டுக்கும் நகர்த்தப்பட வேண்டும்.



படம் 4.13

4.5.6 ஒப்பிடு கட்டளைகள்

(Compare Instructions)

இரு பைட்டுகளையோ அல்லது இரு வார்த்தையையோ ஒப்பிடுவதற்கு, இந்தக் கட்டளைகள் பயன்படுகிறது. மூல இயமி ஒரு பதிவேடாகவோ, ஒரு நினைவக இருப்பிடமாகவோ அல்லது ஒரு எண்ணாகவோ இருக்கலாம். சேர்ப்பிடஇயமி ஒரு பதிவேடாகவோ அல்லது ஒரு நினைவக இருப்பிடமாகவோ இருக்கலாம். மூல இயமியைச் சேர்ப்பிட இயமியிலிருந்து கழித்தல் மூலம் இரு பைட்டுகளை ஒப்பிடலாம். இவ்வாறு கழிப்பதால், மூலம் மற்றும் சேர்ப்பிடங்களின் உள்ளடக்கம் மாறாது ஒப்பிடு கட்டளைகள், பின்வருமாறு அமையும்.

CMP சேர்ப்பிடம், மூலம்

(உ.ம்) CMP CL, BL CMP AX, SI CMP AL, 60H CMP [SI], AL.

CMP CL, BL : CL பைட் மற்றும் BL பைட்டை ஒப்பிட வேண்டும். அதாவது(CL) – (BL). CL மற்றும் BL இன் உள்ளடக்கம் மாறாது.

ஒப்பிடு கட்டளைகள், பொதுவாக நிபந்தனையுடன் கூடிய தாவும் கட்டளைகளுடன் (Conditional Jump Instruction) பயன்படுத்தப்படும்.

4.5.7 தாவும் கட்டளைகள்

(Jump Instructions)

இந்தக் கட்டளைகள் திட்ட நிரலில் உள்ள கட்டளைகளின் இயல்பு வரிசையில் மாற்றம் செய்யும். இந்தக் கட்டளையைச் செயல்படுத்தும்போது, இயல்பு வரிசையில் மாற்றம் ஏற்பட்டுக் கட்டளையில் குறிப்பிடப்பட்டுள்ள முகவரியையுடைய நினைவக இருப்பிடத்திலுள்ள கட்டளை செயல்படுத்தப்படும். இந்தக் கட்டளைகள் இரண்டு வகைப்படும். i) நிபந்தனையற்ற தாவும் கட்டளைகள் ii) நிபந்தனையுடன் கூடிய தாவும் கட்டளைகள்

நிபந்தனையற்ற தாவும் கட்டளைகள்

(Unconditional Jump Instructions)

இந்தக் கட்டளை இரண்டு வகைப்படும். 1) பகுதிக்குள் நடைபெறும் தாவல் (INTRA segment JUMP or NEAR JUMP) 2) பகுதிகளுக்கிடையே நடைபெறும் தாவல் (INTER segment JUMP or FAR JUMP)

பகுதிக்குள் நடைபெறும் தாவல்

இந்த வகையான தாவல், ஒரே குறியீடு பகுதியில் (Code segment) வெவ்வேறு முகவரிகளுக்கிடையே நடைபெறும். இந்த வகையில் IPஇன் மதிப்பு மட்டும் மாறும். CS இன் மதிப்பு மாறாது. இது இரண்டு வகைப்படும்.

1) நேரடித் NEAR தாவல்

(உ.ம்) JMP 1200 H. இந்தக் கட்டளையில் இடப்பெயர்ச்சி முகவரி 1200H, ஒரே குறியீடு பகுதியில் இருக்கும்.

2) மறைமுகத் NEAR தாவல்

(உ.ம்) JMP BX இந்தக் கட்டளையில் இடப்பெயர்ச்சி முகவரி, BX பதிவேட்டில் இருக்கும்.

பகுதிகளுக்கிடையே நடைபெறும் தாவல்

இந்த வகையில் IPஇன் மதிப்பு மற்றும் CSஇன் மதிப்பும் மாறும். இதில் தாவல், நினைவகத்திலுள்ள வெவ்வேறு பகுதிகளுக்கிடையே நடைபெறும். இந்த வகை இரண்டு வகைப்படும்.

1) நேரடித் FAR தாவல்

இந்த வகையில் IPமற்றும் CSஇன் மதிப்பு கட்டளையிலே கொடுக்கப்பட்டிருக்கும்.

2) மறைமுகத் FAR தாவல்

இந்த வகையில், IP மற்றும் CSஇன் மதிப்பு, தரவுப் பகுதியிலுள்ள அடுத்தடுத்த நான்கு நினைவக இருப்பிடங்களில் இருக்கும்.

நிபந்தனையுடன் கூடிய தாவல் கட்டளைகள்

(Conditional Jump Instructions)

இந்த வகையான தாவல், நிலைப்பாட்டுக் கொடியின் நிலையைப் பொறுத்து நடைபெறும். பெரும்பாலான நிபந்தனையுடன் கூடிய தாவல் கட்டளைகளின் செயல்பாடு, ஒரு கொடியின் நிலையைப் பொறுத்து அமையும். ஆனால் சில கட்டளைகளின் செயல்பாடு, ஒரு கொடிக்கு மேலான கொடிகளின் நிலையைப் பொறுத்து அமையும். (உ.ம்)

-) JC கட்டளை, CF இன் நிலையை மட்டும் பார்க்கும். $CF = 1$ ஆக இருந்தால், தாவல் நடைபெறும். $CF = 0$ ஆக இருந்தால், தாவல் கட்டளைக்கு அடுத்துள்ள கட்டளையிலிருந்து செயற்பாடு நடைபெறும். அதாவது தாவல் நடைபெறாது.

) JBE கட்டளையின் செயல்பாடு, CF மற்றும் ZF இன் நிலையை பொறுத்து அமையும். அட்டவணை (4.1) இல், நிபந்தனையுடன் கூடிய தாவல் கட்டளைகள் மற்றும் கொடிகளின் நிலை ஆகியவை தரப்பட்டுள்ளன.

அட்டவணை (4.1)

கட்டளை	நிபந்தனை
JA / JNBE	CF = 0 மற்றும் ZF = 0
JAE / JNB	CF = 0
JB / JNAE	CF = 1
JBE / JNA	CF = 1 or ZF = 1
JC	CF = 1
JE / JZ	ZF = 1
JG / JNLE	ZF = 0 மற்றும் SF = 0
JGE / JNL	SF = 0
JL / JNGE	SF ≠ 0
JLE / JNG	ZF = 1 அல்லது SF ≠ 0
JNC	CF = 0
JNE / JNZ	ZF = 0
JNO	OF = 0
JNS	SF = 0
JNP / JPO	PF = 0
JO	OF = 1
JP / JPE	PF = 1
JS	SF = 1
JCXZ	CX = 0

4.5.8 வளைய கட்டளைகள்

(Loop Instructions)

CX பதிவேட்டின் உள்ளடக்கத்தை ஒரு பிட் குறைத்தல் மற்றும் நிபந்தனையுடன் கூடிய தாவல் ஆகியவற்றின் தொகுப்பே, வளையக் கட்டளையாகும். இந்த கட்டளையைக் கொண்டு, ஒரு குறிப்பிட்ட செயற்பாட்டை பல தடவை செய்யலாம். இந்தக் கட்டளையைச் செயல்படுத்தும்போது, CXஇன் உள்ளடக்கத்தை ஒரு பிட் குறைக்க வேண்டும். மற்றும் CXஇன் மதிப்பு சுழியாகும்வரை, இந்தக் கட்டளையைச் செயல்படுத்த வேண்டும். CXஇன் மதிப்பு சுழியாகிவிட்டால், வளையக் கட்டளைக்கு அடுத்த கட்டளை செயல்படுத்தப்படும். வளையக் கட்டளைக்கான உதாரணம்

1. LOOPZ / LOOPE
2. LOOPNZ / LOOPNE

4.5.9 கம்பிக் கட்டளைகள்

(String Instructions)

கம்பி என்பது, தொடர் தரவு வார்த்தைகள் மற்றும் பைட்டுகளாலானது. இந்தத் தரவு வார்த்தைகள் மற்றும் பைட்டுகள், அடுத்தடுத்துள்ள நான்கு நினைவக இருப்பிடங்களில் சேமிக்கப்படும். இந்தக் கட்டளைகளால், தரவுகளை மூலத்திலிருந்து சேர்ப்பிடத்திற்கு நகர்த்துதல், சேமித்தல், பளுவேற்றுதல், வருடுதல் மற்றும் ஒப்பிடுதல் போன்ற வேலைகளைச் செய்ய முடியும்.

எல்லாக் கம்பி கட்டளைகளுக்கும் நினைவக மூலம் DS : SI என்றும் நினைவகச் சேர்ப்பிடம் ES : DI என்றும் அமையும். அதாவது மூல முகவரி தரவுப் பகுதியிலும், இடப்பெயர்ச்சி முகவரி SI என்ற சுட்டிக்காட்டியிலும் இருக்கும். அதே போன்று சேர்ப்பிட முகவரி கூடுதல் பகுதியிலும், மற்றும் இடப்பெயர்ச்சி முகவரி DI என்ற சுட்டிக்காட்டியிலும் இருக்கும். திசைக்கொடி DF சுழியாக இருந்தால், SI மற்றும் DI சுட்டிக்காட்டிகளின் உள்ளடக்கம் அதிகரிக்கப்படும். திசைக்கொடி DF ஒன்றாக இருந்தால், சுட்டிக்காட்டிகளின் உள்ளடக்கம் குறைக்கப்படும். பைட்டுகளுக்கு அதிகப்படுத்துதல் / குறைத்தல் ஒன்றாக இருக்கும். வார்த்தைகளுக்கு அதிகப்படுத்துதல் / குறைத்தல் இரண்டாக இருக்கும்.

(SI) ← (SI) ± 1 மற்றும் (DI) ← (DI) ± 1(பைட்டுகளுக்கு)

(SI) ← (SI) ± 2 மற்றும் (DI) ← (DI) ± 2 (வார்த்தைகளுக்கு)

DF = 0 ஆக இருந்தால் நேர்க் குறியீடும்

DF = 1 ஆக இருந்தால் எதிர்க் குறியீடும் பயன்படுத்தப்படும்.

நகர்த்துதல் கம்பிக் கட்டளைகள்

(Move String Instruction)

(உ.ம்)

1) MOV SB – கம்பி பைட்டுகளை நகர்த்துதல்

2) MOV SW - கம்பி வார்த்தைகளை நகர்த்துதல்

தரவுப் பகுதியிலுள்ள நினைவக இருப்பிடத்திலிருந்து ஒரு பைட் அல்லது வார்த்தை, கூடுதல் பகுதியிலுள்ள நினைவக இருப்பிடத்திற்கு நகர்த்தப்படும். கூட்டாளர் (assembler)பயன்படுத்தும்போது,MOVS என்ற கம்பிக் கட்டளை பயன்படுத்தப்படும்.

சேமித்தல் கம்பிக் கட்டளைகள்

(Store string Instructions)

(உ.ம்)

1) **STOSB-** கம்பி பைட்டுகளை சேமிக்க வேண்டும். அதாவது ALஇன் உள்ளடக்கத்தை கூடுதல் பகுதியிலுள்ள நினைவக இருப்பிடத்தில் சேமிக்க வேண்டும்.

2) **STOSW**-கம்பி வார்த்தையை சேமிக்க வேண்டும். அதாவது AXன் உள்ளடக்கத்தை கூடுதல் பகுதியில் அடுத்தடுத்துள்ள இரண்டு நினைவக இருப்பிடத்தில் சேமிக்க வேண்டும். கூட்டாளரைப் பயன்படுத்தும்போது, STOS என்ற கம்பிக் கட்டளையைப் பயன்படுத்த வேண்டும்.

பளுவேற்றும் கம்பிக் கட்டளைகள்

(Load string Instructions)

(உ.ம்)

1) LODSB - கம்பி பைட்டைப் பளுவேற்ற வேண்டும்.அதாவது தரவுப் பகுதியிலுள்ள நினைவக இருப்பிடத்திலுள்ள பைட்டை, AL பதிவேட்டில் பளுவேற்ற வேண்டும்.

2) **LODSW** - கம்பி வார்த்தையைப் பளுவேற்ற வேண்டும். அதாவது தரவுப் பகுதியிலுள்ள நினைவக இருப்பிடத்திலுள்ள வார்த்தையை AX பதிவேட்டில் பளுவேற்ற வேண்டும்.கூட்டாளரைப் பயன்படுத்தும்போது, LODS என்ற கம்பி கட்டளையைப் பயன்படுத்த வேண்டும்.

ஒப்பிடு கம்பிக் கட்டளைகள்

(Compare string Instructions)

(உ.ம்)

1) CMPSB கம்பி பைட்டை ஒப்பிடுதல் வேண்டும். அதாவது கம்பி மூலத்திலுள்ள பைட்டைச் சேர்ப்பிடத்திலுள்ள பைட்டுடன் ஒப்பிடுதல் வேண்டும். கழித்தல் மூலம் இரண்டு பைட்டுகள் ஒப்பிடப்படும்.

2) CMPSW- கம்பி வார்த்தைகளை ஒப்பிடுதல் வேண்டும். கூட்டாளரை பயன்படுத்தும் போது CMPS ன்ற கம்பி கட்டளையை பயன்படுத்த வேண்டும்.

வருடுதல் கம்பி கட்டளைகள்

(Scan String Instructions)

(உ.ம்)

1) SCASB - ALஇன் உள்ளடக்கம், கம்பியின் சேர்ப்பிடத்திலுள்ள பைட்டுடன்

ஒப்பிடப்படும்.

2) SCASW – AX இன் உள்ளடக்கம், கம்பியின் சேர்ப்பிடத்திலுள்ள வார்த்தையுடன் ஒப்பிடப்படும். கூட்டாளரைப் பயன்படுத்தும்போது SCAS என்ற கம்பி கட்டளையைப் பயன்படுத்த வேண்டும்.

இதுவரை பார்த்த கம்பிக் கட்டளைகள், ஒரு பைட் மற்றும் ஒரு வார்த்தையைக் கொண்டு செயல்படக் கூடிய கட்டளைகள். இந்தக் கம்பி கட்டளைகளைப் பலமுறை பயன்படுத்த வேண்டுமென்றால், 'மறுபடி செய்' (REPEAT) என்ற சொல், கம்பிக் கட்டளைக்கு முன்னால் வர வேண்டும். CX பதிவேட்டில் உள்ள எண்ணிக்கை, கம்பிக் கட்டளையை எத்தனை தடவை மறுபடி மறுபடி செயல்படுத்த வேண்டும் என்பதைத் தீர்மானிக்கும். ஒவ்வொரு தடவை கம்பிக் கட்டளையைச் செயல்படுத்திய பிறகும், CX இன் மதிப்பு ஒரு பிட் குறைக்கப்படும். REP, REPZ / REPE மற்றும் REPNZ / REPNE போன்றவை, 'மறுபடி செய்' என்றசெயலுக்கான கட்டளைகளாகும்.

4.5.10 கொடி மற்றும் முறைப்படுத்திக் கட்டுப்பாட்டுக் கட்டளைகள் (Flag and Processor Control Instructions)

கொடி கட்டுப்பாட்டு கட்டளைகள்

STC : சுமைக்கொடி 1 என்று அமைவு பெறும்.

CLC : சுமைக்கொடியின் உள்ளடக்கம் சுழியாகும்.

CMC : சுமைக்கொடியின் உள்ளடக்கம் ஒன்றின் நிரப்பு முறையில் மாற்றம் பெறும்.

CLD : திசைக்கொடியின் உள்ளடக்கம் சுழியாகும்.

STD : திசைக்கொடி 1 என்று அமைவு பெறும்

CLI : தடங்கல் செயலியக்கக் கொடியின் உள்ளடக்கம் சுழியாகும்.

STI : தடங்கல் செயலிலக்கக் கொடியின் உள்ளடக்கம் 1 என்று அமைவு பெறும்.

LAHF : கொடிப் பதிவேட்டின் உள்ளடக்கத்தை AH பதிவேட்டில் பளுவேற்ற வேண்டும்.

SAHF : AH இன் உள்ளடக்கம் கொடிப் பதிவேட்டில் சேமிக்கப்படும்.

முறைப்படுத்திக் கட்டுப்பாட்டுக் கட்டளைகள்

HLT : முறைப்படுத்தி, கட்டளைகளைக் கொணர்வது மற்றும் அவற்றைச் செயல்படுத்துவதை நிறுத்திவிடும்.

WAIT : இந்தக் கட்டளையைச் செயற்படுத்தினால், முறைப்படுத்தியானது பிற கட்டளைகளைச் செயல்படுத்துவதை நிறுத்திவிடும். $\overline{TEST}$ முனை ஒன்றாக இருக்கும் வரை 8086, எந்தக் கட்டளையையும் செயல்படுத்தாது. $\overline{TEST}$ முனை சுழிநிலைக்கு வந்துவிட்டால் 8086, கட்டளைகளை செயல்படுத்துவதை தொடங்கிவிடும்.

NOP : 8086 நுண்முறைப்படுத்தியில் எந்த வேலையும் நடைபெறாது.

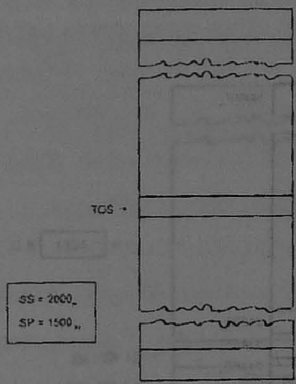
ESC : இந்தக் கட்டளையைச் செயல்படுத்துவதால், உடன் நுண்முறைப்படுத்திக்கு (Coprocessor) பிற கட்டளைகள் அனுப்பப்படும்.

4.5.11 அடுக்கு மற்றும் அடுக்குத் தொடர்பான கட்டளைகள் (Stack and Stack related Instruction)

இன்டெல் 8086 நுண்முறைப்படுத்தி நினைவகத்தின் கொள்ளளவு, 1MB ஆகும். நினைவகம் 16 பகுதியாக பிரிக்கப்பட்டுள்ள. ஒவ்வொரு பகுதியின் கொள்ளளவு 64 KB ஆகும். 8086 நுண்முறைப்படுத்தியில் ஒரே நேரத்தில் நான்கு பகுதிகள் மட்டும் பயன்படுத்தப்படுகின்றன. அந்த நான்கு பகுதிகள் குறியீடு பகுதி, தரவு பகுதி, அடுக்குப் பகுதி மற்றும் கூடுதல் பகுதியாகும். ஒவ்வொரு பகுதியின் தொடக்க முகவரியும் அந்தந்தப் பகுதிப் பதிவேட்டால் தீர்மானிக்கப்படுகிறது.

நினைவகத்தின் அடுக்குப் பகுதி, அடுக்குத் தொடர்பான செயல்பாட்டிற்காகப் பயன்படுகிறது. அடுக்குப் பகுதி பதிவேட்டில் 2000_H என்ற முகவரி இருப்பதாகக் கருதுவோம். அடுக்குப் பகுதியின் தொடக்க முகவரி (SS)0= 20000_H. அடுக்குப் பகுதியின் கொள்ளளவு 64KB ஆக (=$FFFF_H$) இருப்பதால், இந்தப் பகுதியின் கடைசி முகவரி 20000 + FFFF = $2FFFF_H$ ஆகும். அடுக்குத் தொடர்பான எல்லாச் செயல்பாடுகளும், 20000_H முதல் $2FFFF_H$ வரை நடைபெறும். அடுக்குப் பகுதியில் தற்போது பயன்பாட்டிலுள்ள நினைவக இருப்பிடத்தின் முகவரி அடுக்குச் சுட்டிக்காட்டியால் தீர்மானிக்கப்படுகிறது. அதாவது, அடுக்குச்சுட்டிக்காட்டியில் செயலூக்கம் உடைய முகவரி (Effective address or offset address) இருக்கும்.SPஇல் 1500_H என்ற செயலூக்கமுடைய முகவரி இருப்பதாகக் கருதுவோம். நினைவக இருப்பிடத்தின் தரவு சார்ந்த முகவரிPA= $20000_H + 1500_H = 21500_H$.

இந்த இருப்பிடத்தை அடுக்கு உச்சி எனலாம். அடுக்குப் பகுதி படம் (4.14) ல் கார்ட்டப்பட்டுள்ளது. அடுக்குச் சார்ந்த கட்டளைகள் PUSH மற்றும் POP கட்டளையாகும்.



படம் (4.14) அடுக்குப் பகுதி நினைவகம்.

PUSH கட்டளை

பொதுவாக இந்தக் கட்டளையின் அமைப்பு, கீழ்க்கண்டவாறு இருக்கும்.

PUSH மூலஇயமி

இந்தக் கட்டளையிலுள்ள மூல இயமி, ஒரு பொது நோக்கப் பதிவேட்டிலோ, ஒரு பகுதி பதிவேட்டிலோ அல்லது நினைவக இருப்பிடத்திலோ இருக்கும். (உ.ம்) PUSH AX, PUSH DS மற்றும் PUSH [BX] PUSH கட்டளையின் செயல்பாடு பற்றிக் கீழே காண்போம்.

$SS = 2000_H$, $SP = 1500_H$

∴ $PA = 21500_H$

$AX = 5678_H$

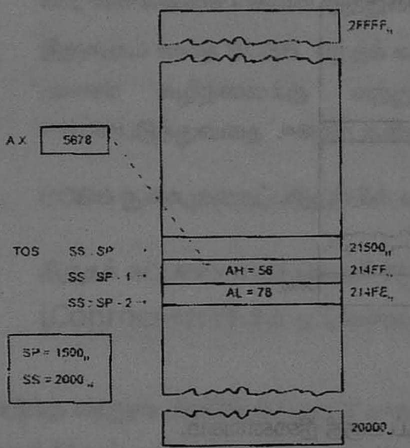
PUSH AXஎன்ற கட்டளையைச் செயல்படுத்தும்போது,AHஇன் உள்ளடக்கம்அதாவது, 56_H ஐ, SP – 1 என்ற முவரியையுடைய அடுக்கு நினைவகத்தில் சேமிக்க வேண்டும். அதே போன்று AL(78_H)இன் உள்ளடக்கத்தை, SP–2 என்ற முகவரியையுடைய அடுக்கு நினைவகத்தில் சேமிக்க வேண்டும். SPஇன் உள்ளடக்கத்தில் இரண்டு பிட் குறைக்க வேண்டும். அதாவது,

$(SS) : (SP) - 1) \leftarrow (AH)$

$(SS) : (SP) - 2) \leftarrow (AL)$

$(SP) : (SP) \leftarrow 2$

இது படம் (4.15)இல் தெளிவாகக் காட்டப்பட்டுள்ளது.



படம் (4.15) AXஇன் உள்ளடக்கத்தை நகர்த்துதல்

POP கட்டளை

இந்தக் கட்டளையின் அமைப்பு, கீழ்க்கண்டவாறு இருக்கும்.

POP சேர்ப்பிட இயமி

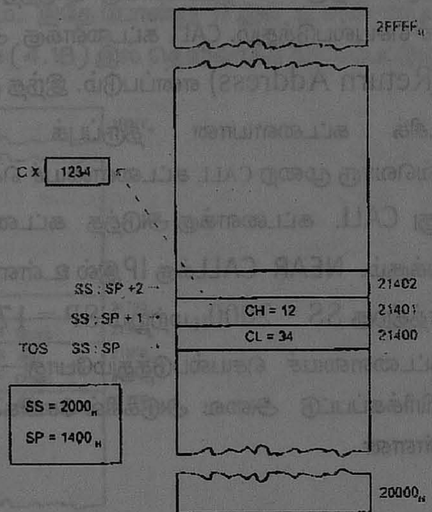
சேர்ப்பிட இயமி, ஒரு பொது நோக்கப் பதிவேடாகவோ, ஒரு பகுதிப் பதிவேடாகவோ (CS)ஐ தவிர அல்லது ஒரு நினைவக இருப்பிடமாகவோ இருக்கலாம். (உ.ம்) POP CX, POP DS மற்றும் POP [SI] POP CX கட்டளையைச் செயல்படுத்துவதாகக் கருதுவோம். இந்தக் கட்டளையைச் செயல்படுத்தும்போது SS : SP என்ற நினைவக இருப்பிடத்திலுள்ள ஒரு பைட்டைCL பதிவேட்டில் சேமிக்க வேண்டும். அதே போன்று SS : SP + 1 என்ற நினைவக இருப்பிடத்திலிருந்து ஒரு பைட்டை CH பதிவேட்டில் சேமிக்க வேண்டும். மற்றும் SPஇன் மதிப்பு இரண்டு பிட் அதிகரிக்கப்படும். அதாவது,

$(CL) \leftarrow (SS : (SP))$

(CH) ⟵ (SS: (SP)+1)

(SP) ⟵ (SP)+2

இது படம் (4.16) இல் தெளிவாகக் காட்டப்பட்டுள்ளது.



படம் (4.16) அடுக்கில் உள்ள CXஇன் உள்ளடக்கத்தை மீண்டும் CX க்கு நகர்த்துதல்

PUSHF / POPF கட்டளைகள்

கொடிப் பதிவேட்டில் உள்ளதைச் சேமிக்க PUSHF என்ற கட்டளை பயன்படுத்தப்படும். நினைவக இருப்பிடங்களிலிருந்து கொடிப் பதிவேட்டில் சேமிக்க POPF என்ற கட்டளை பயன்படுத்தப்படும்.

4.5.12 அழைப்பு மற்றும் திருப்பு கட்டளைகள் (CALL and RET Instructions)

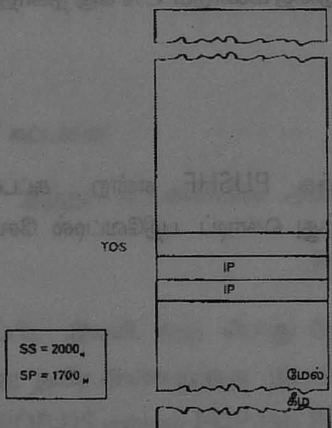
இந்தக் கட்டளைகள் துணை நிரற் செயல்முறையைச் (Subroutine Program) செயல்படுத்தப் பயன்படுகின்றன. CALL கட்டளைகள் இரண்டு வகைப்படும். அவை,

i) பகுதிக்குள் ஏற்படும் அழைப்பு (Intra Segment CALL or Near CALL)

ii) பகுதிகளுக்கிடையே ஏற்படும் அழைப்பு(Inter Segment CALL or Far CALL)

பகுதிக்குள் ஏற்படும் அழைப்பு

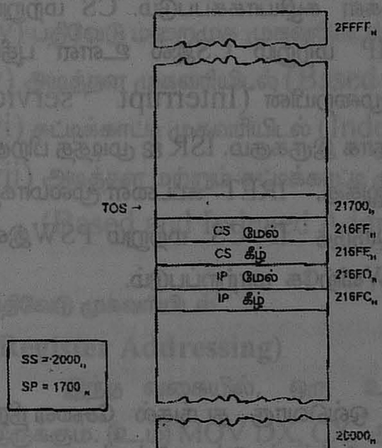
இந்த அழைப்புகளில், துணைநிரற்செயல்முறை ஒரே குறியீடு பகுதியில் இருக்கும். இந்தத் துணைநிரற்செயல்முறையை அழைப்பதற்கு IPஇன் உள்ளடக்கம் மட்டும் மாற்றப்படும். CALL கட்டளையைச் செயல்படுத்திய பிறகு, நுண்முறைப்படுத்தியானது துணைநிரற்செயல்முறையைச் செயல்படுத்தும். துணை நிரற்செயல்முறையைச் செயல்படுத்திய பிறகு நுண்முறைப்படுத்தியானது, முதன்மைத் திட்ட நிரலுக்குத் திரும்பிவிடும். பிறகு CALL கட்டளைக்கு அடுத்த கட்டளையிலிருந்து, முதன்மைத் திட்ட நிரலைச் செயல்படுத்தும். CALL கட்டளைக்கு அடுத்த கட்டளையின் முகவரி, திருப்பு முகவரி (Return Address) எனப்படும். இந்த முகவரி, துணை நிரற் செயல்முறையில் கடைசிக் கட்டளையான திருப்புக் (RET) கட்டளையில் குறிப்பிடப்பட்டிருக்கும். ஒவ்வொரு முறை CALL கட்டளையைச் செயல்படுத்தும்போதும், நுண்முறைப்படுத்தியானது CALL கட்டளைக்கு அடுத்த கட்டளையின் முகவரியை அடுக்கில் சேமித்து வைக்கும். NEAR CALLக்கு IPஇல் உள்ள முகவரி அடுக்கில் சேமிக்கப்படும். உதாரணத்திற்கு $SS = 2000_H$ மற்றும் $SP = 1700_H$ இல் இருப்பதாக கருதுவோம். CALL கட்டளையைச் செயல்படுத்தும்போது IPஇன் உள்ளடக்கம், இரண்டு பைட்டுகளாக பிரிக்கப்பட்டு அவை அடுக்கில் சேமிக்கப்படும். இவை படம் (4.17) இல் காட்டப்பட்டுள்ளன.



படம் (4.17) பகுதிக்குள் நடைபெறும் தாவலின்போது IPஇன் உள்ளடக்கத்தை அடுக்குக்கு நகர்த்துதல்

பகுதிகளுக்கிடையே ஏற்படும் அழைப்பு

இந்த அழைப்புக் கட்டளையைப் பயன்படுத்தும்போது, CS மற்றும் IPஇன் மதிப்பு அடுக்கில் சேமிக்கப்படும். உதாரணத்திற்கு SS = 2000_H மற்றும் SP =1700_H எனக் கருதுவோம். பிறகு CSஇன் பெரும முக்கியத்துவம் வாய்ந்த பைட்டுகள் SP-1 என்ற முகவரியிலும், CSஇன் சிறும முக்கியத்துவம் வாய்ந்த பைட்டுகள் SP-2 என்ற முகவரியிலும் சேமிக்கப்படும். இதே போன்று IPஇல் உள்ள பைட்டுகளும் அடுக்கில் சேமிக்கப்படும். இவை படம் (4.18) இல் தெளிவாகக் காட்டப்பட்டுள்ளன.



படம் (4.18) பகுதிகளுக்கிடையே நடைபெறும் தாவலின்போது CS மற்றும் IP உள்ளடக்கங்களை அடுக்குக்கு நகர்த்துதல்

4.5.13 தடங்கல் தொடர்பான கட்டளைகள்

(Interrupt related Instructions)

8086 நுண்முறைப்படுத்தியின் தடங்கல் தொடர்பான கட்டளைகள் STI, CLI, INTn, IRET மற்றும் INTO ஆகியவையாகும். இவற்றைப் பற்றிய தெளிவான விளக்கத்தைக் கீழே காண்போம்.

STI :

இது அமைவுத் தடங்கல் செயலியக்கக் கொடியாகும். புறச்சைகையை, INTR முனை வழியாக அனுப்பி 8086 நுண்முறைப்படுத்தியில் தடங்கலை ஏற்படுத்தலாம். ஆனால், STI கட்டளையைப் பயன்படுத்தித் தடங்கல் செயலியக்கக் கொடி அமைவு பெற்றால் மட்டுமே, 8086இல் தடங்கலை ஏற்படுத்த முடியும்.

CLI :

இந்தக் கொடியைச் செயல்படுத்தினால், தடங்கல் செயலியக்கக் கொடி மறு அமைவு பெறும். மற்றும் புறத் தடங்கல் உள்ளீடு செயல் முடக்கப்படும்.

INTn :

இது ஒரு மென்பொருள் தடங்கல் கட்டளையாகும். இந்தக் கட்டளையைச் செயல்படுத்தினால், முறைப்படுத்தியானது திட்டநிரலில் உள்ள இயல்பான கட்டளை வரிசையைச் செயல்படுத்துவதை நிறுத்திக்கொள்ளும். கொடிப் பதிவேட்டின் உள்ளடக்கம் அடுக்குக்கு நகர்த்தப்படும். IF மற்றும் TF கொடிகள் சுழியாக்கப்படும். CS மற்றும் IPஇன் உள்ளடக்கம் அடுக்குக்கு நகர்த்தப்படும். IP மற்றும் CSஇல் உள்ள புதிய உள்ளடக்கங்கள், தடங்கல் சேவை நிரற்செயல்முறையின் (Interrupt service routine – ISR) தொடக்க முகவரியைக் குறிப்பதாக இருக்கும். ISR ஐ முடித்த பிறகு, நுண்முறைப்படுத்தியானது முதன்மைத் திட்ட நிரலுக்கு, IRET கட்டளை மூலமாகத் திரும்பிவிடும். அடுக்கில் சேமித்து வைக்கப்பட்டிருந்த IP, CS மற்றும் PSWஇன் உள்ளடக்கங்கள், மறுபடியும் IP, CS மற்றும் PSWவிற்கே மாற்றப்படும்.

IRET :

இது தடங்கல் திருப்புக் கட்டளையாகும். ஒவ்வொரு தடங்கல் சேவை நிரற் செயல்முறை முடிவிலும், IRET கட்டளை சேர்க்கப்படும். இந்தக் கட்டளை செயல்படுத்தப்படும்போது, அடுக்கில் உள்ள IP மற்றும் CS இல் உள்ளடக்கங்கள் மறுபடியும் IP மற்றும் CS க்கு நகர்த்தப்படும்.

INTO :

பிரவாகக் கொடி அமைவு பெற்றால், INTO கட்டளை செயல்படுத்தப்பட்டு 8086 நுண்முறைப்படுத்தியில் INTN கட்டளையில் குறிப்பிடப்பட்டுள்ள செயல்பாடுகள் நடைபெறும்.

4.6 முகவரியிடல் வகைகள்
(Addressing Modes)

இடைநிலை மொழி கட்டளைகள் (Assembly Language Instruction) இயக்க குறியீடு மற்றும் இயமிகளால் ஆனவை. 8086 ஒரு கட்டளையைச் செயல்படுத்தும்போது, ஒரு குறிப்பிட்ட வேலையை இயமிகள் (தரவுகள்) கொண்டு செய்து முடிக்கிறது. இந்த இயமிகள் ஒரு கட்டளையின் ஒரு பகுதியாகவோ, உள்ளார்ந்த பதிவேடுகளிலோ அல்லது நினைவகம் மற்றும் உள்ளீடு/வெளியீடு நிலையங்களிலோ

சேமித்து வைக்கப்பட்டிருக்கும். கட்டளைகளுக்குத் தேவையான தரவுகளைத் தெளிவாகக் குறிப்பிடப் பல யுக்திகள் கையாளப்படுகிறது. இந்த யுக்திகளே, முகவரியிடல் வகைகள் எனப்படுகிறது. இன்டெல் 8086 நுண்முறைப்படுத்தியில் பின்வரும் முகவரியிடல்பயன்படுத்தப்படுகிறது.

i) பதிவேடு முகவரியிடல் (Register Addressing)
ii) உடனடி முகவரியிடல் (Immediate Addressing)
iii) நேரடி முகவரியிடல் (Direct Addressing)
iv) பதிவேடு மறைமுக முகவரியிடல் (Register Indirect Addressing)
v) அடித்தள முகவரியிடல் (Based Addressing)
vi) சுட்டிக்காட்டி முகவரியிடல் (Indexed Addressing)
vii) அடித்தள மற்றும் சுட்டிக்காட்டி முகவரியிடல் (Based and Indexed Addressing)

பதிவேடு முகவரியிடல்

(Register Addressing)

இந்த வகையில், ஒரு உள்ளார்ந்த பதிவேடு, இயமியின் மூலமாக இருக்கும். (உ.ம்) MOV DX, CX. அதாவது CX பதிவேட்டின் உள்ளடக்கத்தை DX பதிவேட்டிற்கு நகர்த்த வேண்டும். இங்கு மூலம் மற்றும் சேர்ப்பிட இயமிகள், பதிவேடுகளின் உள்ளடக்கமாக இருப்பதாகக் குறிப்பிடப்படுகின்றன.

உடனடி முகவரியிடல்

(Immediate Addressing)

இந்த வகையில், கட்டளையிலே இயமி குறிப்பிடப்பட்டிருக்கும். (உ.ம்) MOV BL, 36H. அதாவது 36H என்ற 8-பிட் தரவை BL பதிவேட்டிற்கு நகர்த்த வேண்டும்.

நேரடி முகவரியிடல்

(Direct Addressing)

செயலூாக்கமுடைய முகவரி (Effective Address – EA), ஒரு 16-பிட் எண்ணாகும். நேரடி முகவரியிடலில் இந்த எண் நேரடியாகக் குறிப்பிடப்பட்டிருக்கும். (உ.ம்) MOV BL, [1200H]. இந்தக் கட்டளையில் 1200H என்பது, தரவுப் பகுதி அடித்தளத்திலிருந்து இடம் பெயர்ந்த முகவரியாகும். அதாவது தரவுப் பகுதியில், தரவு சார்ந்த முகவரி (Physical address) என்பது, PA = (DS)0 + 1200H ஆகும்.

DS:1200H என்ற நினைவக இருப்பிடத்தில், 36_H என்ற எண் இருப்பதாக கருதுவோம். மேலே குறிப்பிடப்பட்ட கட்டளையின்படி, 36_H என்ற எண்ணை BL பதிவேட்டிற்கு நகர்த்த வேண்டும்.

பதிவேடு மறைமுக முகவரியிடல்
(Register Indirect Addressing)

இந்த முகவரியிடல் வகையில், செயலூக்கமுடைய முகவரி (Effective Address) சுட்டிக்காட்டிப் பதிவேட்டிலோ (SI or DI) அல்லது அடித்தளப் பதிவேட்டிலோ (BX) அல்லது அடித்தளச் சுட்டிக்காட்டியிலோ (BP) இருக்கும். ஒவ்வொன்றுக்கும் தரவு சார்ந்த முகவரி (Physical address), பின்வருமாறு பெறப்படுகிறது.

PA = (DS)0 + (SI) → DS:SI
(or) PA = (DS)0 + (DI) → DS:DI
(or) PA = (DS)0 + (BX) → DS:BX
(or) PA = (SS)0 + (BP) → SS:BP

(உ.ம்) MOV AX, [SI]

DS : SI என்ற நினைவக இருப்பிடத்தின் உள்ளடக்கத்தை AL பதிவேட்டிற்கும் மற்றும் DS : SI+1என்ற நினைவக இருப்பிடத்தின் உள்ளடக்கதை AH பதிவேட்டிற்கும் நகர்த்த வேண்டும்.

அடித்தள முகவரியிடல்
(Based Addressing)

இந்த வகையில், ஒரு இயமியின் செயலூக்கமுடைய முகவரி என்பது 8-பிட் அல்லது 16-பிட் இடப்பெயர்ச்சி முகவரி மற்றும் தரவுப் பகுதியிலுள்ள BX பதிவேட்டின் உள்ளடக்கத்தின் கூட்டுத்தொகை ஆகும். அல்லது அடுக்குப் பகுதியிலுள்ள BP பதிவேட்டின் உள்ளடக்கம் மற்றும் 8-பிட் அல்லது 16-பிட் இடப்பெயர்ச்சி முகவரியின் கூட்டுத்தொகையாகும். (உ.ம்) MOV AL, SUM(BX)

EA = SUM + (BX) மற்றும் PA = (DS)0 + SUM + (BX)

சுட்டிக்காட்டி முகவரியிடல்
(Indexed Addressing)

இந்த வகை முகவரியிடல், அடித்தள முகவரியிடல் போன்றது. ஆனால் இந்த வகையில், ஒரு இயமியின் செயலூக்கமுடைய முகவரி (EA) என்பது 8-பிட் அல்லது 16-பிட் இடப்பெயர்ச்சி முகவரி மற்றும் தரவுப் பகுதியிலுள்ள SI மற்றும் DI பதிவேட்டின் உள்ளடக்கத்தின் கூட்டுத்தொகையாகும்.(உ.ம்) MOV AL, ARRAY [SI] இநதக் கட்டளைப்படி EA = ARRAY + (SI) மற்றும் PA = (DS)0 + ARRAY + (SI).

அடித்தள மற்றும் சுட்டிக்காட்டி முகவரியிடல்

(Based and Indexed Addressing)

இந்த வகையில் செயலூாக்கமுடைய முகவரி (EA) என்பது, அடித்தளப் பதிவேடு (BX அல்லது BP) மற்றும் சுட்டிக்காட்டிப் பதிவேடு (SI அல்லது DI) ஆகியவற்றின் கூட்டுத்தொகையாகும். (உ.ம்) MOV AL, [BX] [SI] இந்தக் கட்டளைப்படி EA = (BX) + (SI) மற்றும் PA = (DS)0 + (BX) + (SI), கட்டளையில் கூடுதலாக இடப்பெயர்ச்சி முகவரி சேர்க்கப்பட்டால், பின் அந்த வகைமுகவரியிடலை அடித்தள மற்றும் சுட்டிக்காட்டியைச் சார்ந்த முகவரியிடல் வகை (Relative Based and Indexed Addressing Mode) எனலாம்.

(உ.ம்) MOV AL, SUM [BX] [SI]

இதில் EA = SUM + BX + SI

PA = (DS)0 + EA

4.7 8086 நுண்முறைப்படுத்தியின் தடங்கல் அமைப்பு

(Interrupt Structure of 8086 Microprocessor)

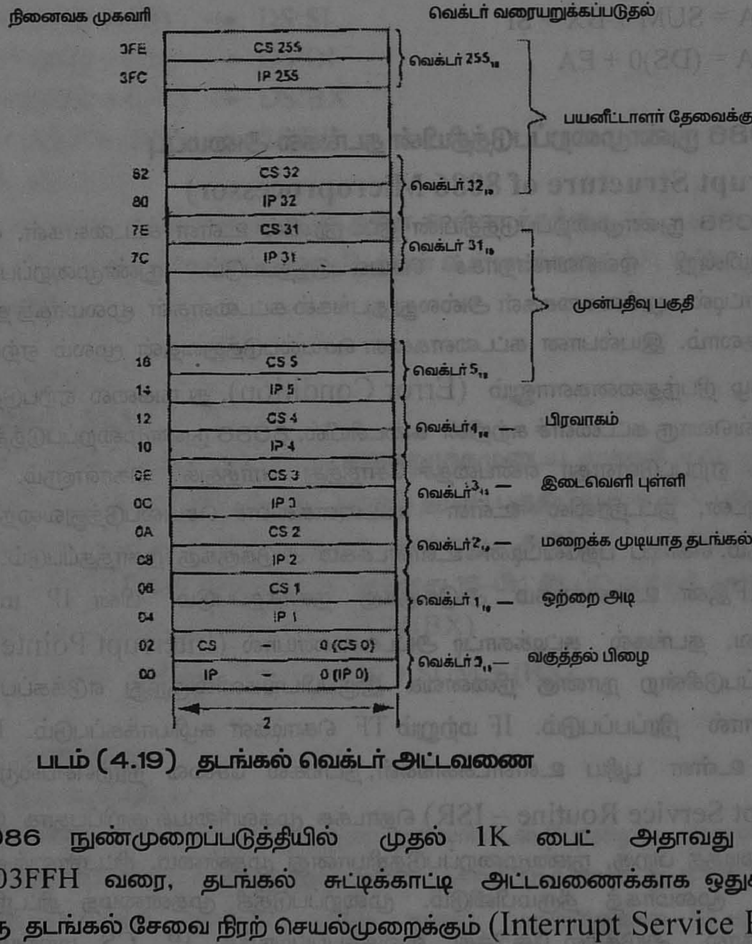
8086 நுண்முறைப்படுத்தியின் திட்ட நிரலில் உள்ள கட்டளைகள், எந்தவிதத் தடங்கலுமின்றி ஒவ்வொன்றாகச் செயல்படுத்தப்படும். நுண்முறைப்படுத்தியின் செயல்பாட்டில், புறச்சைகைகள் அல்லது தடங்கல் கட்டளைகள் மூலமாகத் தடங்கலை ஏற்படுத்தலாம். இயல்பான கட்டளைகளை செயல்படுத்துவதன் மூலம் ஏற்படக்கூடிய சில பிழை நிபந்தனைகளாலும் (Error Condition), தடங்கலை ஏற்படுத்தலாம்.

ஒவ்வொரு கட்டளைச் சுற்றின் கடைசியில், 8086 நுண்முறைப்படுத்தி ஏதாவது தடங்கல் ஏற்பட்டுள்ளதா என்பதைச் சோதித்து பார்த்துக் கொள்ளும். தடங்கல் ஏற்பட்டவுடன், திட்டநிரலில் உள்ள கட்டளைகளைச் செயல்படுத்துவதை நிறுத்திக் கொள்ளும். கொடிப் பதிவேட்டின் உள்ளடக்கம் அடுக்குக்கு நகர்த்தப்படும். பின் CS மற்றும் IPஇன் உள்ளடக்கம் அடுக்குக்கு நகர்த்தப்படும். பின் IP மற்றும் CS ஆகியவை, தடங்கல் சுட்டிக்காட்டி அட்டவணையால் (Interrupt Pointer Table) குறிப்பிடப்படுகின்ற நான்கு நினைவக இருப்பிடங்களிலிருந்து எடுக்கப்பட்ட புதிய மதிப்புகளால் நிரப்பப்படும். IF மற்றும் TF கொடிகள் சுழியாக்கப்படும். IP மற்றும் CS இல் உள்ள புதிய உள்ளடக்கங்கள், தடங்கல் சேவை நிரற்செயல்முறையின் (Interrupt Service Routine – ISR) தொடக்க முகவரியைக் குறிப்பதாக இருக்கும். ISR ஐ முடித்த பிறகு, நுண்முறைப்படுத்தியானது முதன்மை திட்டநிரலுக்கு IRET கட்டளை மூலமாகத் திரும்பிவிடும். முறைப்படுத்தி முதன்மைத் திட்டநிரலுக்குத் திரும்புவதால், அடுக்கில் சேமித்து வைக்கப்பட்டிருந்த IP, CS மற்றும் கொடிப் பதிவேட்டின் உள்ளடக்கங்கள், மறுபடியும் அவற்றிற்கே மாற்றப்படும்.

4.7.1 தடங்கல் சுட்டிக்காட்டி அட்டவணை

(Interrupt Pointer Table)

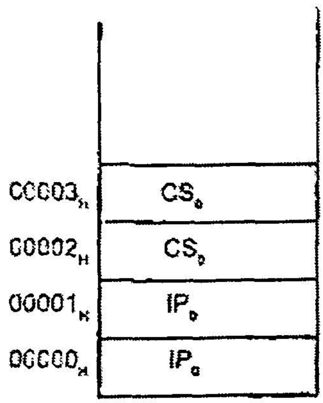
தடங்கல் சேவை நிரற் செயல்முறையின் (Interrupt Service Routine) தொடக்க முகவரி, தடங்கல் சுட்டிக்காட்டி (Interrupt Pointer) அல்லது தடங்கல் வெக்டர் (Interrupt Vector) எனப்படும். தடங்கல் சுட்டிக்காட்டி அட்டவணையானது, தடங்கல் வகை எண்ணையும் (Interrupt type number), அவற்றின் நிரற் செயல்முறையின் இருப்பிடங்களோடு தொடர்புபடுத்தி காட்டுகிறது.படம் (4.19)இல் சுட்டிக்காட்டி அட்டவணையின் வரைபடம் காட்டப்பட்டுள்ளது. இந்த அட்டவணையில், 256 முகவரி சுட்டிக்காட்டிகள் உள்ளன.



படம் (4.19) தடங்கல் வெக்டர் அட்டவணை

8086 நுண்முறைப்படுத்தியில் முதல் 1K பைட் அதாவது 00000H முதல் 003FFH வரை, தடங்கல் சுட்டிக்காட்டி அட்டவணைக்காக ஒதுக்கப்படும். ஒவ்வொரு தடங்கல் சேவை நிரற் செயல்முறைக்கும் (Interrupt Service Routine) தேவையான CS மற்றும் IPஇன் மதிப்புகள் நான்கு பைட் நினைவக இருப்பிடங்களில் சேமிக்கப்படும். எனவே, 256 தடங்கல் செயல்முறைக்கு 1024 நினைவக

இருப்பிடங்கள் (256 X 4 =1024) தேவைப்படுகின்றன. IPஇன் மதிப்பு, வெக்டரின் கீழ் வார்த்தைகளாகச் (Low word) சேமிக்கப்படும். Sஇன் மதிப்பு வெக்டரின் மேல் வார்த்தைகளாகச் (High word) சேமிக்கப்படும். உதாரணத்திற்கு INTO என்ற தடங்கல் வகைக்கு IPஇன் மதிப்பு 00000H மற்றும் 00001லும், CSஇன் மதிப்பு 00002H மற்றும் 00003Hலும் சேமிக்கப்படும். இவை படம் (4.20)இல் காட்டப்பட்டுள்ளன. 00004H முதல் 00007 H வரையிலான நான்கு இருப்பிடங்கள் INT1 என்ற கட்டளை வகைக்காகப் பயன்படுத்தப்பகின்றன. இவ்வாறு மற்ற தடங்கல் வகைகளான INT3 மற்றும் INT4 ஆகியவற்றிற்கு, எட்டு நினைவக இருப்பிடங்கள் ஒதுக்கப்பட்டுள்ளன.



படம் (4.20) INTO தடங்கலின் முகவரி அட்டவணை

INTO, INTI, INT3 மற்றும் INT4 ஆகியவை உள்ளார்ந்த தடங்கல்களாகும். இவற்றை முறையே வகுத்தல் பிழை (Divide error), ஒற்றை அடி (single step). இடைவெளிப் புள்ளி (Break point) மற்றும் பிரவாகம் (over flow) எனவும் அழைக்கலாம். இரண்டாவது சுட்டிக்காட்டி, மறைக்க முடியாத தடங்கலின் (Non maskable interrupt – NMI) நிரற் செயல்முறைகளுக்கான தொடக்க முகவரியை அடையாளம் காட்டும். அடுத்த 27 சுட்டிக்காட்டிகள் (5 முதல் 31 வரை), அட்டவணையின் முன்பதிவு பகுதியை (Reserved Portion) குறிக்கின்றன. மீதமுள்ள 224 சுட்டிக்காட்டிகள் (INT32 முதல் INT255 வரை அதாவது $INT20_H$ முதல் $INTFF_H$ வரை), வன்பொருள் அல்லது மென்பொருள் தடங்கல்களின் தேவைக்காகப் பயன்படுத்தப்படுகிறது.

4.7.2 வன்பொருள் தடங்கல்
(Hardware Interrupt)

புறசாதனங்களிலிருந்து சைகைகளை அனுப்பி,8086 நுண்முறைப்படுத்தியில் தடங்கலை ஏற்படுத்தலாம். இந்தச் சைகைகளை, INTR முனை (முனை18) அல்லது NMI முனையின் (முனை 17) வழியாக அனுப்பலாம். இவை வன்பொருள் தடங்கல்கள் எனப்படும்.

INTR :

இந்தத் தடங்கல் சைகையானது, புறச்சாதனங்களுக்கு ஊர்திச் சேவை தேவைப்படுகிறது என்பதை, 8086 நுண்முறைப்படுத்திக்கு உணர்த்தும். ஒவ்வொரு கட்டளைச் சுற்றின் கடைசி கடிகாரச் சுற்றில், INTR சைகை வந்திருக்கிறதா என்பதை 8086 சோதித்துப் பார்க்கும். INTRசைகையை ஏற்றுக் கொண்டதற்கு அடையாளமாக 8086 நுண்முறைப்படுத்தியானது, தாழ்நிலை $\overline{\text{INTA}}$ சைகையை முனை 24இன் வழியாகப் புறச்சாதனங்களுக்குத் திரும்ப அனுப்பும். $\overline{\text{INTA}}$ சைகையைப் பெற்றுக் கொண்டு புற மின்சுற்றானது, தடங்கல் வகை எண்ணைத் தரவு ஊர்தி வழியாக 8086க்கு அனுப்பும். தடங்கல் வகை எண்ணை 8086 பெற்றுக் கொண்டவுடன், கொடிப் பதிவேட்டில் உள்ளதை அடுக்குக்கு நகர்த்தும். பின் IF மற்றும் TFஐ சுழியாக்கும். CS மற்றும் IPஇன் உள்ளடக்கங்கள் அடுக்குக்கு நகர்த்தப்படும். தடங்கல் சேவை நிரற் செயல்முறைக்கு, தடங்கல் சுட்டிக்காட்டி அட்டவணையிலிருந்து பெறப்பட்ட மதிப்புகளால், IP மற்றும் CS நிரப்பப்படும். நிரற் செயல்முறையின் முடிவில், IRET கட்டளையைச் செயல்படுத்திக் கட்டுப்பாடானது முதன்மைத் திட்டநிரலுக்கு மாற்றப்படும்.

மறைக்க முடியாத தடங்கல் - வகை 2
(Non maskable Interrupt – Type 2)

இதுவும் ஒரு வன்பொருள் தடங்கலாகும். மறைக்க முடியாத தடங்கல் முனையின் வழியாக (முனை 17) உயர்நிலைச் சைகை கிடைக்கப்பெற்றால், 8086 நுண்முறைப்படுத்தி வகை - 2 (INT2) தடங்கலை ஏற்படுத்தும். அதாவது, கொடிப் பதிவேட்டின் உள்ளடக்கம் அடுக்குக்கு நகர்த்தப்படும். IF மற்றும் TFஇன் மதிப்பு, சுழியாகும். மேலும் CS மற்றும் IPஇன் உள்ளடக்கங்கள் அடுக்குக்கு நகர்த்தப்படும். பிறகு தடங்கல் நிரற் செயல்முறைக்காக CS மற்றும் IPஇன் உள்ளடக்கங்கள் தடங்கல் சுட்டிக்காட்டி அட்டவணையிலிருந்து பெறப்பட்ட மதிப்புகளால் நிரப்பப்படும். இந்தத் தடங்கலின் மூலம், மின்சாரம் இல்லாத நிலையைத் தெரிந்து கொள்ளலாம்.

4.7.3 உள்ளார்ந்த தடங்கல்கள்
(Internal Interrupts)

INT0, INT1, INT3 மற்றும் INT4 ஆகியவை, 8086 நுண்முறைப்படுத்தியின் உள்ளார்ந்த தடங்கல்களாகும். மற்ற தடங்கல்களைக் காட்டிலும் இந்த நான்கு தடங்கல்களுக்கு, அதிக முக்கியத்துவம் கொடுக்கப்படுகிறது. இந்த நான்கு தடங்கல்களையும், தடங்கல் செயலியக்கக் கொடியால் மறைக்க முடியாது. உள்ளார்ந்த நான்கு செயல்பாடுகள் i) வகுத்தல் பிழை (Divide Error) ii) ஒற்றை அடி (Single Step) iii) இடைவெளிப் புள்ளி (Break Point) iv)பிரவாகம் (Overflow) ஆகியவையாகும். இந்த நான்கு செயற்பாட்டில் ஏதாவது ஒன்று ஏற்பட்டாலும், 8086 நுண்முறைப்படுத்தியானது, தன்னிச்சையாகத் தெரிந்து கொள்ளும். பின் கட்டுப்பாடானது சேவை செயல்முறைக்கு மாற்றப்படும். இவற்றின் செயல்பாடுகளைப் பற்றிக் கீழே காண்போம்.

வகுத்தல் பிழை தடங்கல் (INT0)

8086 நுண்முறைப்படுத்தி, DIV கட்டளையையோ அல்லது IDIV கட்டளையையோ செயல்படுத்தும்போது, INT0 தடங்கலானது பிழை நிபந்தனையைக் குறிக்கும். வகுக்கும் எண் சுழியாக இருந்தாலோ அல்லது DIV அல்லது IDIV கட்டளையைச் செயற்படுத்தும்போது வரக்கூடிய முடிவு சேருமிடத்தில் சேமிக்க முடியாத அளவுக்குப் பெரிதாக இருந்தாலோ வகுத்தல் பிழை ஏற்படும். வகுத்தல் பிழை கண்டுபிடிக்கப்பட்டால், 8086 நுண்முறைப்படுத்தியானது INT0 தடங்கலைச் செய்யும். இதனால் 8086ல் வழக்கமாக ஏற்படக்கூடிய மாற்றங்கள் ஏற்படும்.

ஒற்றை அடி தடங்கல் (INT1)

பொறிக் கொடி (Trap flag) அமைவு பெறும்போது, ஒற்றை அடி முறை செயற்பாடு செயலியக்கம் பெறும். பொறிக் கொடி ஒன்று என அமைவு பெற்றால் 8086 முறைப்படுத்தியானது, ஒவ்வொரு கட்டளையையும் செயல்படுத்திய பிறகு INT1 தடங்கலை ஏற்படுத்தும். பின் 8086இல் வழக்கமாக ஏற்படக்கூடிய மாற்றங்கள் ஏற்படும்.

இடைவெளிப் புள்ளி தடங்கல் (INT3)

INT3 தடங்கலை ஒரு கட்டளையாகவும் பயன்படுத்திக் கொள்ளலாம். இந்த கட்டளையைத் திட்டநிரலில் இடையில் சேர்த்துக் கொள்ளலாம். இவ்வாறு சேர்த்தால், முறைப்படுத்தியானது திட்டநிரலில் INT3க்கு முந்திய கட்டளைகளைச் செயல்படுத்தும். INT3 வந்தவுடன், இடைவெளிப்புள்ளி செயல்முறையைச் செயல்படுத்தத் தொடங்கிவிடும்.

பிரவாகத் தடங்கல் (INT4)

இரு குறியீடு எண்களைக் கொண்டு நடந்த கணிதச் செயல்பாடுகளின் முடிவு, சேருமிடப் பதிவேட்டிலோ அல்லது நினைவக இருப்பிடங்களிலோ சேமிக்க முடியாத அளவு பெரிதாக இருந்தால், பிரவாகக் கொடி ஒன்று என அமைவு பெறும். இவ்வாறு அமைவு பெறுவதன் மூலம், பிரவாகநிலையை அது உணர்த்துகிறது. பிறகுமுறைப்படுத்தியானது INT4 தடங்கலை ஏற்படுத்தும்.

4.7.4 மென்பொருள் தடங்கல் (வகை 0 முதல் 255 வரை) (Software Interrupts – Types 0 through 255)

8086 நுண்முறைப்படுத்தியில் 256 மென்பொருள் தடங்கல்கள் உள்ளன. இந்த வகையில், மென்பொருள் கட்டளையான INTn ஐ கொண்டு, முறைப்படுத்தியில் தடங்கல் ஏற்படுத்தப்படுகிறது. இதில் n என்பது வகை எண்ணைக் குறிக்கும். இந்த வகைத் தடங்கல்களை தடங்கல் செயலியக்கக் கொடி மூலம் மறைக்க முடியாது. INTn கட்டளையைச் செயல்படுத்தினால் கொடிப் பதிவேட்டின் உள்ளடக்கம் அடுக்குக்கு நகர்த்தப்படும். IF மற்றும் TFஇன் மதிப்பு சுழியாகும். மேலும் CS மற்றும் IPஇன் மதிப்புகள், அடுக்குக்கு நகர்த்தப்படும். பின் கட்டுப்பாடு சேவை, செயல்முறைக்கு மாற்றப்படும். வெளிபுறத் தடங்கல்களை விட, இவை அதிக முக்கியதுதுவம் வாய்ந்தவை.

4.8 இன்டெல் 8086இன் திட்ட நிரலாக்கம் (Programming of Intel 8086)

ஒரு குறிப்பிட்ட வேலையை முழுமையாகச் செய்து முடிப்பதற்கான கட்டளைத் தொடர், திட்டநிரல் (Program) எனப்படும். நாம் எழுதிய திட்ட நிரலை, நுண்முறைப்படுத்தி மூலம் செயல்படுத்தி பார்ப்பதற்கு, செயற்குறிப்பு (Mnemonics) மூலம் குறிப்பிடப்பட்ட கட்டளைகளை இயந்திர குறியீடாக மாற்ற வேண்டும். ஒவ்வொரு கட்டளையிலும் இரண்டு பகுதிகள் இருக்கும். ஒன்று இயக்கக் குறியீடு (Opcode) மற்றொன்று இயமிகள் (Operands) இன்டெல் 8086 கட்டளைகளின் நீளம் 1 முதல் 6 பைட்டுகள் வரை இருக்கும். இதில் முதல் பைட், எப்பொழுதும் இயக்கக் குறியீடாக இருக்கும். சில கட்டளைகளின் இயந்திரக் குறியீடுகள், ஒன்று அல்லது பல பைட்டுகளால் ஆனவை. இந்தப் பைட்டுகளையும் இயக்கக் குறியீடுகள் எனலாம்.

இன்டெல் 8085 நுண்முறைப்படுத்தியில் 246 கட்டளைகள் உள்ளன. அவற்றிற்கான இயக்கக் குறியீடுகள், உடனடி அட்டவணையில் (Readymade chart) கொடுக்கப்பட்டுள்ளன. அந்த மாதிரி உடனடி அட்டவணை 8086 நுண்முறைப்படுத்திக்கு கிடையாது. அந்த அட்டவணையைத் தயார் செய்வதும் கடினமானது. ஏனென்றால், MOV கட்டளைக்கு 2^{10}(=1024) விதமான இயக்கக் குறியீடுகள் இருக்கும். இதே போன்று ADD, SUB மற்றும் சில கட்டளைகளுக்கு

நிறைய இயக்கக் குறியீடுகள் உள்ளன. எனவே, கட்டளை மாதிரிகளைப் பயன்படுத்தி (Instruction template), இயந்திரக் குறியீடுகளைத் திட்டமிடுபவரே (Programmer) உருவாக்க வேண்டும். இவ்வாறு இயந்திரக் குறியீடுகளை உருவாக்குவது மிகவும் கடினமானது. இதனைத் தவிர்க்கக் 'கூட்டாளர்' (Assembler) என்ற மென்பொருளைக் கொண்டும் இயந்திரக் குறியீடுகளை உருவாக்கலாம்.

4.8.1 MOV instruction

(Template for MOV instruction)

பின்வருமாறு, மூலத்திலிருந்து சேர்ப்பிடத்திற்கு இயமியை நகர்த்தக்கூடிய MOV கட்டளைக்கான குறியீடு மாதிரி (Coding template), அட்டவணை (4.2) இல் கொடுக்கப்பட்டுள்ளது.

a) ஒரு பதிவேட்டிலிருந்து மற்றொரு பதிவேட்டிற்கு இயமியை நகர்த்துதல்

அல்லது

b) ஒரு பதிவேட்டிலிருந்து நினைவக இருப்பிடத்திற்கு இயமியை நகர்த்துதல்.

அல்லது

c) ஒரு நினைவக இருப்பிடத்திலிருந்து பதிவேட்டிற்கு இயமியை நகர்த்துதல்.

மேலே கூறிய மூன்று செயல்களிலும், இயமிகளில் ஒரு இயமி பதிவேடாக இருக்கும்.

பைட் 1	பைட் 2	பைட் 3	பைட் 4
100010 d w	Mod reg r/m	disp. low	disp. high
	அல்லது	addr. low	addr. high

அட்டவணை (4.2)

பைட் 1ல் w = 0 ⟶ பைட் செயற்பாட்டிற்காகப் பயன்படுத்தப்படுகிறது.

w = 1 ⟶ வார்த்தை செயற்பாட்டிற்காக பயன்படுத்தப்படுகிறது.

d = 0 ⟶ பைட் 2இல் உள்ள REG மூலப் பதிவேடாகும்.

d = 1 ⟶ பைட் 2இல் உள்ள REG சேர்ப்பிடப் பதிவேடாகும்.

அட்டவணை (4.3)இல் காட்டியுள்ளபடி பைட் 2இல் உள்ள 3பிட் REG புலம், ஒரு 8பிட்

அல்லது 16பிட் பதிவேட்டைக் குறிக்கிறது. பைட் 2இல், மூன்று பிட்டுகள் REG புலத்தை குறிக்கின்றன. மீதமுள்ள 5பிட்டுகள், பதிவேடு / நினைவகத்தை 32 விதமான வழிகளில் குறிக்கின்றன. 5பிட் புலம், 2பிட் MOD புலமாகவும், ஒரு 3-பிட் R/M புலமாகவும் பிரிக்கப்படுகிறது.

பைட் 2இல் உள்ள 3பிட் REG புலம்

REG	வார்த்தை	பைட்
000	AX	AL
001	CX	CL
010	DX	DL
011	BX	BL
100	SP	AH
101	BP	CH
110	SI	DH
111	DI	BH

அட்டவணை (4.3)

MOD புலமானது, நான்கு விதமான செயற்பாட்டினை முடிவு செய்கிறது (MOD = 00,01,10 மற்றும் 11). ஒவ்வொரு முறைக்கும், 3பிட் R/M புலமானது, இயமிகளில் மற்றொரு இயமியின் பதிவேட்டையோ அல்லது முகவரியிடல் முறையையோ குறிக்கிறது.

MOD 11:

இந்த முறையில், இரண்டு இயமிகளும் பதிவேடுகளில் இருக்கும். இது பதிவேடு முகவரியிடல் முறையைச் சார்ந்தது. அட்டவணை (4.4) இல் கொடுக்கப்பட்டுள்ளபடி, 3பிட் R/M புலமும் ஒரு பதிவேடாகும்.

3பிட் R/M புலம்

R/M	வார்த்தை	பைட்
000	AX	AL
001	CX	CL
010	DX	DL
011	BX	BL
100	SP	AH
101	BP	CH
110	SI	DH
111	DI	BH

அட்டவணை (4.4)

MOD 00

இந்த முறையில் இரண்டாவது இயமி, ஒரு நினைவக இருப்பிடத்தின் உள்ளடக்கமாக இருக்கும். இவை அட்டவணை (4.5)இல் கொடுக்கப்பட்டுள்ளது.

3பிட் **R/M** புலம்

R/M	செயலூாக்கமுடைய முகவரி [EA]	முகவரியிடல் முறை
000	[BX] + [SI]	அடித்தள மற்றும் சுட்டிக்காட்டி
001	[BX] + [DI]	அடித்தள மற்றும் சுட்டிக்காட்டி
010	[BP] + [SI]	அடித்தள மற்றும் சுட்டிக்காட்டி
011	[BP] + [DI]	அடித்தள மற்றும் சுட்டிக்காட்டி
100	[SI]	பதிவேடு மறைமுக முகவரியிடல்
101	[DI]	பதிவேடு மறைமுக முகவரியிடல்
110	கட்டளையின் ஒரு பகுதி	நேரடி முகவரியிடல்
111	[BX]	பதிவேடு மறைமுக முகவரியிடல்

அட்டவணை (4.5)

MOD 01

இந்த முறையில் இரண்டாவது இயமி, நினைவக இருப்பிடத்தில் இருக்கும். ஒரு 8பிட் இடப்பெயர்ச்சி முகவரி, செயலூாக்கம் உடைய முகவரி கணக்கிடுதலில் சேர்த்துக் கொள்ளப்படும். இவை அட்டவணை (4.6)இல் கொடுக்கப்பட்டுள்ளன.

3-பிட் R/M புலம்

R/M	செயலூாக்கமுடைய முகவரி [EA]	முகவரியிடல் முறை
000	[BX] + [SI] + d8	அடித்தள மற்றும் மறைமுக முகவரியிடல்
001	[BX] + [DI] + d8	அடித்தள மற்றும் மறைமுக முகவரியிடல்
010	[BP] + [SI] + d8	அடித்தள மற்றும் மறைமுக முகவரியிடல்
011	[BP] + [DI] + d8	அடித்தள மற்றும் மறைமுக முகவரியிடல்
100	[SI] + d8	மறைமுக முகவரியிடல்
101	[DI] + d8	மறைமுக முகவரியிடல்
110	[BP] + d8	அடித்தள முகவரியிடல்
111	[BX] + d8	அடித்தள முகவரியிடல்

அட்டவணை (4.6)

MOD 10

இந்த முறையில் இரண்டாவது இயமி, நினைவக இருப்பிடத்தில் இருக்கும். ஒரு 16பிட் இடப்பெயர்ச்சி முகவரி, செயலூக்கம் உடைய முகவரி கணக்கிடுதலில் சேர்த்துக் கொள்ளப்படும். இவை அட்டவணை (4.7)இல் கொடுக்கப்பட்டுள்ளன.

3-பிட் R/M புலம்

R/M	செயலூக்கமுடைய முகவரி [EA]	முகவரியிடல் முறை
000	[BX] + [SI] + d16	அடித்தள மற்றும் மறைமுக முகவரியிடல்
001	[BX] + [DI] + d16	அடித்தள மற்றும் மறைமுக முகவரியிடல்
010	[BP] + [SI] + d16	அடித்தள மற்றும் மறைமுக முகவரியிடல்
011	[BP] + [DI] + d16	அடித்தள மற்றும் மறைமுக முகவரியிடல்
100	[SI] + d16	சுட்டிக்காட்டி முகவரியிடல்
101	[DI] + d16	சுட்டிக்காட்டி முகவரியிடல்
110	[BP] + d16	அடித்தள முகவரியிடல்
111	[BX] + d16	அடித்தள முகவரியிடல்

அட்டவணை (4.7)

மேலே குறிப்பிடப்பட்டுள்ள அட்டவணைகளைப் பயன்படுத்தி, சில கட்டளைகளுக்கு இயந்திரக் குறியீடுகளை எவ்வாறு பெறலாம் என்பதைப் பார்ப்போம்.

1) தரவுகளை ஒரு பதிவேட்டிலிருந்து மற்றொரு பதிவேட்டிற்கு நகர்த்துதல்

MOV AX, BX

AX மற்றும் BX ஆகியவை 16-பிட் பதிவேடுகள் .

இந்த கட்டளைக்கான மாதிரி

பைட் -1	பைட் -2
100010 D W	MOD REG

மேலே கொடுக்கப்பட்டுள்ள கட்டளைப்படி, தரவை ஒரு 16-பிட் பதிவேட்டிலிருந்து மற்றொரு 16-பிட் பதிவேட்டிற்கு நகர்த்த வேண்டும். அதனால் MOD = 11.

மேலும் வார்த்தைச் செயல்பாட்டிற்கு W = 1 D=0 என்றால், பைட்-2இல் உள்ள REG என்பது மூல பதிவேடாக இருக்கும். REG = BX =011.R/M என்பதும் ஒரு பதிவேடாகும். இந்தக் கட்டளைப்படி R/M என்பது AX = 000. எனவே, இரண்டு பைட் குறியீடுகள்

இயக்கக் குறியீடு d w	mod reg r/m
100010 0 1	11 011 000
89	d8

எனவே MOV AX, BX என்ற கட்டளையின் இயக்கக் குறியீடு 89, D8 ஆகும்.

இதே கட்டளைக்கு D=1 என்றால், பைட் -2 இல் உள்ள REG ஒரு சேர்ப்பிடப் பதிவேடாகும். அதனால் AX = 000. R/M என்பதும் ஒரு பதிவேடாகும். R/M என்பது BX = 011

இயக்கக் குறியீடு	d	w	mod	reg	r/m
100010	1	1	11	000	011

8B C3

MOV AX, BX என்ற கட்டளையின் இரண்டாவது குறியீடு 8B, C3 எனவே, இந்த வகையான MOV கட்டளைக்கு இரண்டு இயந்திரக் குறியீடுகள் உள்ளன. இவற்றில் ஏதாவது ஒன்றைப் பயன்படுத்திக் கொள்ளலாம்.

2) தரவை ஒரு நினைவக இருப்பிடத்திலிருந்து ஒரு பதிவேட்டிற்கு நகர்த்துதல்
MOV AX, [SI]

பைட் 2இல் உள்ள REG ஒரு சேர்ப்பிடப் பதிவேட்டைக் குறிக்கிறது. எனவே, AX = 000 மற்றும் D=1, W=1. SI பதிவேட்டின் உள்ளடக்கம் செயலூக்கமுடைய முகவரியாகும். R/M என்பது SI பதிவேடாகும். MOD 00 என்ற முறையில் SI=100

இயக்கக் குறியீடு	d	w	mod	reg	r/m
100010	1	1	00	000	100

8B 04

MOV AX, [SI] யின் இயந்திரக் குறியீடு 8B, 04 ஆகும்.

3) உடனடி தரவை ஒரு பதிவேட்டிற்கு நகர்த்துதல்
MOV AL, 25
இந்தக் கட்டளைக்கான மாதிரி

1011	W	reg	d8

W=0, reg = AL = 000, d8 = 25H

இயக்கக் குறியீடு w reg d8

1011 0 000	0010 0101
B0	25

MOV AL, 25 இன் இயந்திரக் குறியீடு BO, 25_H ஆகும்.

4) இரண்டு பதிவேடுகளின் உள்ளடக்கத்தைக் கூட்டுதல்

ADD AL, BL

இந்தக் கட்டளைகளுக்கான மாதிரி

000000 d w	mod reg r/m

d=0 எனில் பைட் 2-இல் உள்ள reg மூலமாகச் செயல்படும். எனவே BL=011,W=0, MOD=11 மற்றும் r/m = AL = 000

000000 0 0	11 011 000
00	D8

ADD, AL, BL இன் இயந்திரக் குறியீடு 00, $D8_H$ ஆகும்.

4.8.2 பகுதிப் புறக்கணிப்பு முற்சேர்க்கை (Segment override prefix)

குறியீடு பகுதியிலிருந்து (Code segment) கட்டளை கொணர்தல் நடைபெறும்போது, செயலூக்கமுடைய முகவரி (EA), IP இல் இருக்கும். அடுக்குப் பகுதியில் அடுக்குச் செயற்பாட்டிற்கான செயலூக்கமுடைய முகவரி, SPஇல் இருக்கும். பதிவேடு மறைமுக முகவரியிடல் முறையில், தரவுப் பகுதியிலுள்ள செயலூக்கமுடைய முகவரி BX, SI அல்லது DIஇல் இருக்கும். அதே போன்று அடுக்கு பகுதியிலுள்ள செயலூக்கமுடைய முகவரி BPஇல் இருக்கும். CSஇல் உள்ள செயலூக்கமுடைய முகவரி BX பதிவேட்டிலும், DS உள்ள செயலூக்கமுடைய முகவரியை BP பதிவேட்டிலும் வைக்க வேண்டுமென்றால், பகுதிப் புறக்கணிப்பு முற்சேற்க்கையைப் (Segment override prefix) பயன்படுத்த வேண்டும். உதாரணத்திற்கு MOV[BX], AL இந்தக் கட்டளைப்படி ALஇல் உள்ளதை DS உள்ள நினைவக இருப்பிடத்திற்கு நகர்த்த வேண்டும். செயலூக்கமுடைய முகவரி BXஇல் இருக்கும்.

ஆனால் ALஇல் உள்ளதை CSஇல் உள்ள நினைவக இருப்பிடத்திற்கு நகர்த்த வேண்டுமென்றால், பகுதிப் புறக்கணிப்பு முற்சேர்க்கையைப் பயன்படுத்த வேண்டும். எனவே மேலே குறிப்பிடப்பட்ட கட்டளை கீழ்கண்டவாறு சிறிது மாற்றி அமைக்கப்படும்.

MOV CS : [BX], AL

இந்தக் கட்டளைப்படி செயலூக்கமுடைய முகவரி BX இல் இருக்கும். பகுதிப் புறக்கணிப்பு முற்சோ;க்கை, பயன்படுத்தக்கூடிய கட்டளைகளின் இயந்திர குறியீட்டை உருவாக்கக் கூடுதல் பைட் தேவைப்படும். அந்தப் பைட்டைப் பகுதிப் புறக்கணிப்பு முற்சேர்க்கை பைட் எனலாம். (Segment override prefix byte – SOP byte). SOP பைட்டின் அமைப்பு, கீழ்க்கண்டவாறு இருக்கும்.

001	SR	110

SR இன் மதிப்பு அட்டவணை (4.8) இல் கொடுக்கப்பட்டள்ளது.

SR	பகுதி பதிவேடு
00	ES
01	CS
10	SS
11	DS

அட்டவணை (4.8)

4.8.3 கூட்டாளர் மற்றும் கூட்டாளர் வழிமுறைகள் (Assembler and Assembler Directives)

கூட்டாளர் (Assembler) என்பது, ஒரு மென்பொருளாகும். இந்த மென்பொருள், இடை மொழியில் (Assembly Language) எழுதப்பட்ட கட்டளைகளை இயந்திரக் குறியீடாகவோ அல்லது ஈரடிக் குறியீடாகவோ மாற்றம் செய்யும். கூட்டாளர் வழிமுறைகள் கட்டளைகளைப் போன்றவை. இவற்றை கூட்டாளர் மென்பொருளில் மட்டும்தான் பயன்படுத்த முடியும். நுண்முறைப்படுத்தியில் பயன்படுத்த முடியாது. ASSUME, DB, DW, SEGMENT, ENDS, PROC, END போன்றவை சில கூட்டாளர் வழிமுறைகளாகும். இவற்றின் பொருள் மற்றும் செயல்முறைகள் பற்றிக் கீழே காண்போம்.

1) ASSUME

இந்த வழிமுறை, எந்தப் பகுதியைப் (Segment) பயன்படுத்த வேண்டும் என்று கூட்டாளருக்குக் கட்டளையிடும்.

ASSUME CS : C_HERE

இந்த வாக்கியமானது, ஒரு திட்ட நிரலில் உள்ள கட்டளைகள் C_HERE என்று குறிப்பிடப்பட்ட பகுதியில் இருப்பதாகக் கூட்டாளருக்கு உணர்த்தும். தரவுப் பகுதியைப் (Data segment – DS) பயன்படுத்த வேண்டுமென்றால் D_HERE

என்றும், அடுக்குப் பகுதியைப் (Stack segment – SS) பயன்படுத்த வேண்டுமென்றால் S_HERE என்றும், கூட்டாளர் வழிமுறையில் குறிப்பிட வேண்டும்.

ASSUME DS : D_HERE

ASSUME SS : S_HERE

2) DB - பைட்டை வரையறுத்தல்

இந்த வழிமுறையின்படி நினைவகத்தில் ஒரு இருப்பிடமோ அல்லது பல இருப்பிடங்களோ தரவுகளைச் சேமிக்க ஒதுக்கப்படும்.

NUM1 DB 45H

இந்த வாக்கியப்படி, NUM1 என்ற மாறிக்காக (Variable), ஒரு பைட் நினைவக இருப்பிடம் ஒதுக்கி வைக்கப்படும் மற்றும் 45H என்ற மதிப்பு, அந்த நினைவக இருப்பிடத்தில் சேமிக்கப்படும். உதாரணத்திற்கு,

MOV AL, NUM1

கட்டளைப்படி தரவுப் பகுதியிலுள்ள NUM1 என்று குறிப்பிடப்பட்ட நினைவக இருப்பிடத்தில் உள்ளதை AL க்கு நகர்த்த வேண்டும்.

3) DW - வார்த்தையை வரையறுத்தல்

இந்த வழிமுறையானது, ஒரு வார்த்தையைச் சேமிக்க, அடுத்தடுத்து உள்ள இரண்டு நினைவக இருப்பிடத்தை ஒதுக்குமாறு கூட்டாளருக்குக் கட்டளையிடும்.

MULTIPLIER DW 1234H என்ற வாக்கியப்படி MULTIPLIER என்று குறிப்பிடப்பட்ட நினைவக இருப்பிடத்தில் 1234H என்ற தரவு சேமிக்கப்படும்.

4) SEGMENT - தொடக்கப் பகுதி

இந்த வழிமுறையானது, தொடக்கப் பகுதி எது என்பதைக் கூட்டாளருக்கு உணர்த்தும்.

(உ.ம்) C_HERE SEGMENT

இந்த வாக்கியமானது, தொடக்கப் பகுதி C_HERE என்பதைக் கூட்டாளருக்கு உணர்த்துகிறது.

5) ENDS - முடிவு பகுதி

இந்த வழிமுறையானது, முடிவு பகுதி எது என்பதைக் கூட்டாளருக்கு உணர்த்தும்.

6) PROC - செயல்முறை

இந்த வழிமுறை, துணைத் திட்ட நிரலின் தொடக்கம் எது என்பதைக் கூட்டாளருக்கு உணர்த்தும்.

7) EQU

இந்த வழிமுறை திட்டநிரலில் வரக்கூடிய ஒரு மதிப்பிற்கோ அல்லது குறியீட்டிற்கோ பெயர் சூட்டப் பயன்படுத்தப்படும்.

(உ.ம்) VALUE EQU 39H

8) END - திட்ட நிரலின் முடிவு

இந்த வழிமுறை, ஒரு திட்டநிரலின் கடைசி வாக்கியமாக அமையும். இந்த வாக்கியத்தின் மூலம் திட்டநிரல் முடிந்து விட்டது என்பது, கூட்டாளருக்குத் தெரிவிக்கப்படும்.

சாதாரணமாக 8086 நுண்முறைப்படுத்தியில், MASM (கூட்டாளர்) என்ற மென்பொருளைக் கொண்டு திட்டநிரல் செயல்படுத்தப்படும். திட்டநிரலை செயல்படுத்த கூட்டாளரைத் தவிர EDITOR மற்றும் LINKER என்ற மென்பொருள்களும் தேவைப்படுகின்றன. EDITOR மென்பொருளைக் கொண்டு, இடைநிலை மொழியில் எழுதப்பட்ட திட்டநிரலை உடைய ஒரு கோப்பை (File) உருவாக்கலாம். திட்டநிரல்கள், வன்வட்டுகளில் (Hard Disk) சேமித்து வைக்கப்படுகின்றன.

4.8.4 திட்டநிரல் அமைப்பு
(Program Structure)

முதலில் திட்டநிரலுக்குப் பெயர் கொடுக்க வேண்டும். பெயரின் அளவு 8 எழுத்துகளுக்கு (Character) மேல் மிகாமல் இருக்க வேண்டும். இந்தக் கோப்பைச் (file) சேமிக்கும்போது, ASM என்று பெயருக்கு பின்னால் நீட்டிப்புச் செய்ய வேண்டும். பெயருக்கு அடுத்த வரி அரைப்புள்ளியுடன் (semi colon) தொடங்கும். இந்த வரி ஒரு குறிப்பை உணர்த்தும். கூட்டாளர், இந்தக் குறிப்பிற்காக எந்தக் குறியீடையும் உருவாக்காது. பின், தரவு பகுதி D_HERE என பெயரிடப்பட்டிருக்கும். இந்தப் பகுதியில் திட்டநிரலில் வரக்கூடிய மாறிலி மற்றும் மாறி (Variable) ஆகியவை வரையறுக்கப்பட்டிருக்கும். இதற்காக EQU, DB, DW போன்ற கூட்டாளர் வழிமுறைகள் பயன்படுத்தப்படும் D_HERE SEGMENT மற்றும் D_HERE ENDS ஆகியவை, தரவுப் பகுதியின் தொடக்கம் மற்றும் முடிவாக இருக்கும். இடைநிலை

மொழியில் எழுதப்பட்ட கட்டளைகள், குறியீடு பகுதியில் இருக்கும். குறியீடு பகுதி C_HERE எனப் பெயரிடப்படும். C_HERE SEGMENT மற்றும் C_HERE ENDS ஆகியவை, குறியீடு பகுதியின் தொடக்கம் மற்றும் முடிவாக இருக்கும்.

குறியீடு பகுதியின் முதல் வரி, நிறுத்துக்குறியுடன் கூடிய 'START' என்ற லேபிலைக்(LABEL) கொண்டிருக்கும். குறியீடு பகுதியின் கடைசி வரி END வாக்கியமாக இருக்கும். கட்டளைகள் அனைத்தும் குறியீடு பகுதியில் இருக்கும். JUMP மற்றும் CALL கட்டளைக்கு லேபில் பயன்படுத்தப்படும். எல்லாத் திட்டநிரல்களிலும் முதல் இரண்டு வரிகள், தரவுப் பகுதிப் பதிவேட்டின் உள்ளடக்கத்தை முடிவு செய்வதற்காக பயன்படுத்தப்படும். தரவுப் பகுதியின் தொடக்கமுகவரியின் மேல் 16 - பிட்டுகள் DS பதிவேட்டில் இருக்கும். திட்டநிரலில் உள்ள D_HERE என்பவை இந்த 16-பிட் முகவரியைக் குறிக்கும். தரவை நேரடியாக எந்தவொரு பகுதிப் பதிவேட்டிற்கும் நகர்த்த முடியாது. எனவே இந்த மதிப்பு, முதலில் AX பதிவேட்டிற்கும் பின் AX லிருந்து DS பதிவேட்டிற்கும் நகர்த்தப்படும்.

திட்டநிரல், இடைநிலை மொழியில் எழுதப்பட்டு, அவை பின் EDITOR என்ற மென்பொருள் மூலமாக பதிவுப் செய்யப்படும். திட்டநிரலானது ADBYT.ASM என்று சேமிக்கப்படும். இந்தத் திட்டநிரலை நேரடியாகச் செயல்படுத்த முடியாது. எனவே, திட்டநிரலை இயந்திரக் குறியீடுகளாக மாற்றி, பின் அவற்றைச் செயல்படுத்த வேண்டும். பின்வரும் பிரிவுகளில் சில திட்ட நிரல்களைப் பற்றி பார்ப்போம்.

1) இரண்டு 16-பிட் எண்களைக் கூட்டுதல்

(Addition of two 16-bit numbers)

2567H மற்றும் 3689H என்ற இரு எண்களைக் கருதுவோம். EQU வழிமுறையின் மூலம், இந்த எண்கள் NUM1 மற்றும் NUM2 எனப் பெயரிடப்படும். DW என்ற வழிமுறையின் மூலம், இந்த இரு எண்களின் கூட்டுத் தொகையைச் சேமிக்க ஒரு வார்த்தை இருப்பிடம் ஒதுக்கப்படும். இந்த இருப்பிடத்திற்கு RESULT என்று பெயரிடப்படும்.

கட்டளைகள் குறியீடு பகுதியில் இருக்கும். முதல் எண் AX பதிவேட்டிற்கும், இரண்டாவது எண் BX பதிவேட்டிற்கும் நகர்த்தப்படும். ADD AX, BX என்ற கட்டளை, இரு எண்களைக் கூட்டி, முடிவினை AX பதிவேட்டில் சேமிக்கும். பின் முடிவு நினைவகத்திற்கு மாற்றப்படும். மேலே குறிப்பிடப்பட்ட இரு எண்களின் கூட்டுத் தொகை (5BF0)H, தரவுப் பகுதியில் உள்ள வார்த்தை இருப்பிடத்தில் இருக்கும்.

PROGRAM 1

```
NAME        ADDWRDS.ASM
; Adds two 16-bit numbers (WORDS) named NUM1 AND NUM2.

D_HERE      SEGMENT
NUM1        EQU         2567H
NUM2        EQU         3689H
RESULT      DW          1 DUP ( 0 )
D_HERE      ENDS

C_HERE      SEGMENT
            ASSUME      CS : C_HERE,        DS : D_HERE
START:      MOV         AX, D_HERE
            MOV         DS, AX
            MOV         AX, NUM1
            MOV         BX, NUM2
            ADD         AX, BX
            MOV         RESULT, AX
            INT         3
C_HERE      ENDS
            END         START
```

2) இரண்டு நினைவக இருப்பிடங்களில் உள்ள 16-பிட் எண்ணைக் கூட்டுதல்

(Addition of two 16-bit numbers in two memory location)

2567H மற்றும் 3689H என்ற இரு NUM1 மற்றும் NUM2 என்று குறிப்பிடப்பட்ட இரண்டு நினைவக இருப்பிடங்களில், DW என்ற வழிமுறையைப் பயன்படுத்திச் சேமித்து வைக்கப்பட்டுள்ளதாக கருதுவோம். இதற்கான திட்டநிரல் கீழே கொடுக்கப்பட்டுள்ளது. இந்தத் திட்டநிரலுக்கு ADMWRDS என்று பெயரிடப்பட்டுள்ளது. கூட்டாளரானது, இந்தத் திட்டநிரலுக்கு ஆறு நினைவக இருப்பிடங்களை 2567H, 3689H மற்றும் இரண்டு எண்களின் கூட்டுத்தொகையான 5BF0H ஆகியவற்றிற்காக ஒதுக்குகிறது.

PROGRAM 2 :

```
NAME            ADMWRDS .ASM
;Adds two wards stored in memory locations NUM1 and NUM2

D_HERE    SEGMENT
NUM1      DW        2567H
NUM2      DW        3589H
RESULT    DW        1 DUP
D_HERE    ENDS

C_HERE    SEGMENT
          ASSUME CS : C_HERE,  DS : D_HERE

START     MOV  AX, D_HERE
          MOV  DS, AX
          MOV  AX, NUM1
          MOV  BX, NUM2
          ADD  AX, BX
          MOV  RESULT, AX
          INT  3
C_HERE    ENDS
          END  START
```

3) இரண்டு 16-பிட் எண்களைக் கழித்தல்

(Subtraction of two 16-bit numbers)

5BFOH மற்றும் 3689H என்ற இரு 16-பிட் எண்களைக் கருதுவோம். EQU வழிமுறையின் மூலம், மேலே குறிப்பிடப்பட்ட எண்களுக்கு முறையே NUM1 மற்றும் NUM2 என்று பெயர் கொடுப்போம். DW வழிமுறையின் மூலம், கழித்தல் மதிப்பைச் சேமிக்க, நினைவக இருப்பிடங்கள் ஒதுக்கப்படுகின்றன. 5BFOH என்ற எண் AX பதிவேட்டிற்கும், 3689H என்ற எண் BX பதிவேட்டிற்கும் நகர்த்தப்படும். இரண்டு எண்களைக் கழிப்பதற்கு SUB AX, BX என்ற கட்டளை பயன்படுத்தப்படும். இந்தக் கட்டளைப்படி இரண்டு எண்கள் கழிக்கப்பட்டு முடிவு AX இல் சேமித்து வைக்கப்படும். பின் AXஇல் உள்ளவை நினைவகத்திற்கு நகர்த்தப்படும். இதற்கான திட்டநிரல் கீழே கொடுக்கப்பட்டுள்ளது.

PROGRAM 3

```
NAME            SUBWRDS . ASM
; Adds two 16-bit numbers (WORDS) named NUM1 AND NUM2.

D_HERE   SEGMENT
NUM1     EQU       5BF0H
NUM2     EQU       3689H
RESULT   DW        1 DUP ( 0 )
D_HERE   ENDS

C_HERE   SEGMENT
         ASSUME    CS : C_HERE,    DS : D_HERE
START:   MOV       AX, D_HERE
         MOV       DS, AX
         MOV       AX, NUM1
         MOV       BX, NUM2
         SUB       AX, BX
         MOV       RESULT, AX
         INT       3
C_HERE   ENDS
         END       START
```

4) இரண்டு 16–பிட் எண்களைப் பெருக்குதல்

(Multiplication of two 16-bit numbers)

1234H மற்றும் 4567H என்ற இரு எண்களைக் கருதுவோம். இவை வார்த்தை இருப்பிடங்களில் (Word Locations) சேமித்து வைக்கப்படும். MUL என்ற கட்டளை இரு எண்களைப் பெருக்கி, 32 – பிட் முடிவைக் கொடுக்கும். இரு எண்களின் பெருக்கல் மதிப்பு $(04EF\ \ 56EC)_H$, முதல் 16 பிட்டுகள் DX பதிவேட்டிலும், மற்றும் கடைசி 16–பிட்டுகள் AX பதிவேட்டிலும் இருக்கும். பின் AX இல் உள்ளவை PRODUCT மற்றும் PRODUCT+1 என்று குறிப்பிடப்பட்ட நினைவக இருப்பிடங்களில் சேமிக்கப்படும். அதே போன்று DXஇல் உள்ளவை PRODUCT+2 மற்றும் PRODUCT+3 என்று குறிப்பிடப்பட்ட நினைவக இருப்பிடங்களில் சேமிக்கப்படும்.

PROGRAM 4

NAME WRDMUL.ASM ; Multiplies two 16bit numbers and gets a 32bit result

```
D_HERE              SEGMENT
MULTIPLICAND DW             1234H
MULTIPLIER    DW        4567H
PRODUCT       DW        2 DUP ( 0 )
D_HERE        ENDS

C_HERE        SEGMENT
              ASSUME CS ; C_HERE,      DS ; D_HERE
START ;       MOV AX, D_HERE
              MOV DS, AX
              MOV AX, MULTIPLICAND
              MOV BX, MULTIPLIER
              MUL BX
              MOV PRODUCT, AX
              MOV PRODUCT+2, DX
              INT  3
C_HERE        ENDS
              END   START
```

5) 32-பிட் எண்ணை 16-பிட் எண்ணால் வகுத்தல்

(Division of 32-bit number by 16-bt number)

32-பிட் எண் வகுக்கப்படும் எண் (Dividend). இந்த எண், நினைவகத்தின் 4-பைட் இருப்பிடங்களில் சேமிக்கப்படும். இதை DD வழிமுறை கொண்டு செய்ய முடியும். இரட்டை வார்த்தையானது, DX மற்றும் AX பதிவேட்டில் வைக்கப்படும். DXஇல் மேல் வார்த்தையும், AXஇல் கீழ் வார்த்தையும் இருக்கும். 16-பிட் எண் வகுக்கும் எண்ணாகச் (Divisor)செயல்படும். இது இரண்டு பைட் இருப்பிடங்களில் சேமிக்கப்படும். வகுக்கும் எண் BX ல் வைக்கப்படும். இரு எண்களை வகுக்க DIV BX என்ற கட்டளை பயன்படுத்தப்படும். இந்தக் கட்டளையை செயல்படுத்தினால் 16-பிட் ஈவு கிடைக்கும்.இது AX இல் இருக்கும். மற்றும் 16-பிட் மீதி DXஇல் இருக்கும்.

PROGRAM 5

```
NAME WRDDIV . ASM ; Divides a 32-bit number by a 16-bit number.
D_HERE                  SEGMENT
DIVIDEND      DD        12345678H
DIVISOR       DW        0ABCDH
QUOTIENT      DW        1 DUP (0)
REMANIDER     DW        1 DUP (0)
D_HERE        ENDS

C_HERE        SEGMENT
              ASSUME CS : C_HERE, DS : D_HERE
START :       MOV  AX, D_HERE
              MOV  DS, AX
              MOV AX, DIVIDEND
              MOV DX, DIVIDEND+2  ; 1234H in DX
              DIV   DIVISOR

              MOV QUOTIENT, AX
              MOV REMAINDER, DX

              INT    3
     C_HERE   ENDS
              END START
```

6) பல் பைட் கூட்டல்

(Multi byte Addition)

இந்தத் திட்ட நிரலில், ஒரு 8 பைட் எண் மற்றொரு 8 பைட் எண்ணோடு கூட்டப்படும். முடிவும் ஒரு 8 பைட் எண்ணாக இருக்கும். முதல் 8 பைட் எண் நினைவகத்தின் முதல் வரிசையில் (NUMS1), அடுத்தடுத்து வரக்கூடிய 8 நினைவக இருப்பிடத்தில் இருக்கும். இதற்கான வாக்கியம்,

NUMS1, DB 90H, 91H, 92H, 93H, 94H, 95H, 96H, 97H.

90H ⟶ பெரும முக்கியத்துவம் வாயந்த பிட்

97H ⟶ சிறும முக்கியத்துவம் வாய்ந்த பிட்

இரண்டாவது 8 பைட் எண், நினைவகத்தின் இரண்டாவது வரிசையில் (NUMS2) அடுத்தடுத்து வரக்கூடிய 8 நினைவக இருப்பிடத்தில் இருக்கும். கூட்டல் ஒவ்வொரு

பைட்டாக நடக்கும். முதல் எண்ணின் சிறும முக்கியத்துவம் வாய்ந்த பிட் (least significant bit– LSB) ALஇல் வைக்கப்படும். பின் ALஇன் உள்ளடக்கம், நினைவகத்திலுள்ள இரண்டாவது எண்ணின் LSBயுடன் கூட்டப்படும். பின் முடிவு நினைவகத்திற்கு நகர்த்தப்படும். இதே முறையில், மற்ற பைட்டுகளையும் கூட்ட வேண்டும். LSB தவிர மற்ற பைட்டுகளை கூட்டும்போது சுமை கொடியையும் சேர்த்துக் கொள்ள வேண்டும். CX பதிவேடு எண்ணியாகச் செயல்படும். இந்த திட்டநிரலில் CX=8 (8 பைட்). BXஇல் ஏழு என்ற எண் LSB ஐ குறிக்கும். MOV AL, NUMS1 [BX] என்ற கட்டளைப்படி, தரவுப் பகுதியிலுள்ள நினைவகத்தின் உள்ளடக்கம் AL பதிவேட்டிற்கு நகர்த்தப்படும்.

செயலூக்கம் உடைய முகவரி

EA = NUMS1 + (BX)

நினைவகத்தின் தரவு சார்ந்த முகவரி

PA = (DS)0 + NUMS1 + (BX)

இதே முறையில் ADD AL, NUMS2 [BX] மற்றும் MOV SUM [BX], AL கட்டளைகளும் செயல்படுத்தப்படும். CXஇன் உள்ளடக்கம் சுழியாகும் வரை, ஒவ்வொரு பைட்டாகக் கூட்டல் நடைபெறும்.

PROGRAM 6

```
NAME        MBYTADD . ASM ; ADDS A 8-BYTE NUMBER WITH ANOTHER
8-BYTE NUMBER

D_HERE      SEGMENT
NUMS1       DB   90H, 91H, 92H, 93H, 94H, 95H, 96H, 97H
            DB   8 DUP (OFFH)
NUMS2       DB   50H, 61H, 72H, 83H, 94H, 0A5H, 0B6H, 0C7H
            DB   8 DUP (OFFH)
SUM         DB   8 DUP (0)
            DB   8 DUP (OFFH)
D_HERE      ENDS

C_HERE      SEGMENT
            ASSUME CS : C_HERE,   DS ; D_HERE
START       MOV  AX, D_HERE
            MOV  DS, AX
            CLC
            MOV  CX, 8
```

```
                    MOV  BX, 7
L1                  MOV  AL, NUMS1 [BX]
                    ADC  AL, NUMS2 [BX]
                    MOV  SUM [BX], AL
                    DEC  BX
                    LOOP L1
                    INT 3
C_HERE              ENDS
                    END  START
                    END  START
```

7) ஏறுவரிசை

(Ascending order)

அடுத்தடுத்துள்ள நினைவக இருப்பிடங்களில், எட்டு எண்கள் இருப்பதாகக் கருதுவோம். முதல் நினைவக இருப்பிடம் ARRAY எனக் குறிப்பிடப்படுகிறது. வரிசையில் உள்ள முதல் எண் ALக்கு நகர்த்தப்படும். இந்த எண், வரிசையிலுள்ள அடுத்த எண்ணுடன் ஒப்பிடப்படும். முதல் எண் சிறியதாக இருந்தால், அதன் இருப்பிடம் மாற்றப்படாது. அதே சமயம், முதல் எண் இரண்டாவது எண்ணை விடப் பெரியதாக இருந்தால், நினைவகத்தில் இந்த இரண்டு எண்களின் இருப்பிடங்கள் ஒன்றுக்கொன்று மாற்றப்படும். இதே முறை பிற எண்களுக்கும் (அதாவது 2 மற்றும் 3, 3 மற்றும் 4, 4 மற்றும் 5.....) கடைபிடிக்கப்படும். வரிசையில் எட்டு எண்கள் இருந்தால் ஏழு முறை ஒப்பிடுதல் நடைபெறும். முடிவில் மிகப் பெரிய எண், வரிசையில் உள்ள நினைவக இருப்பிடங்களில் முதல் இருப்பிடத்தில் வைக்கப்படும். இதே முறையில் மற்ற எண்களை ஏறுவரிசையில் வரிசைப்படுத்த, மேலே குறிப்பிடப்பட்ட செயல்முறையைத் திரும்பத் திரும்பச் செயல்படுத்த வேண்டும். ஒவ்வொரு முறையும், ஒப்பிடுதலின் எண்ணிக்கை 6, 5, 4 எனக் குறைத்துக் கொண்டே வரும். இதற்கான திட்டநிரல் கீழே கொடுக்கப்பட்டுள்ளது. CX பதிவேடு ஒரு எண்ணியாகச் செயல்படுகிறது. திரும்பத் திரும்பச் செய்யக்கூடிய செயல்பாட்டிற்கு LOOP கட்டளை பயன்படுத்தப்படும்.

உள்ளீடு : 33H, 22H, 99H, 0EEH, 11H, 55H, 44H, 88H.

வெளியீடு : 11, 22, 33, 44, 55, 88, 99, EE

PROGRAM-7

```
NAME       NUMSORTASC . ASM
; To arrange a set of numbers in ascending order

D_HERE     SEGMENT
ARRAY      DB   33H, 22H, 99H, 0EEH, 11H, 55H, 44H, 88H
COUNT      EQU  8
D_HERE     ENDS

C_HERE     SEGMENT
           ASSUME CS: C_HERE, DS:D_HERE
START:     MOV  AX, D_HERE
           MOV  DS, AX
           MOV  BX, COUNT

           DEC  BX

DO_2 :     MOV CX, BX
           MOV SI, 0
DO_1       MOV  AL, ARRAY [SI]
           INC  SI
           CMP  AL, ARRAY [SI]
           JB   AHEAD
           XCHG AL, ARRAY [SI]
           MOV  ARRAY [SI-1],AL
AHEAD :    LOOP DO_1
           DEC  BX
           JNZ  DO_2
           INT  3
C_HERE     ENDS
           END  START
```

8) இறங்கு வரிசை

(Descending order)

கொடுக்கப்பட்ட எண்களை இறங்கு வரிசையில் வரிசைப்படுத்த ஏறுவரிசைக்கான திட்டநிரலைப் பயன்படுத்தலாம். ஆனால் அதில், JB என்ற கட்டளைக்குப் பதிலாக JA என்ற கட்டளைப் பயன்படுத்தப்படும்.

உள்ளீடு : 33H, 22H, 99H, 0EEH, 11H, 55H, 44H, 88H.

வெளியீடு : EE, 99, 88, 55, 44, 33, 22, 11

PROGRAM-8

```
NAME        NUMSORTDSC . ASM
; To arrange a set of numbers in descending order

D_HERE      SEGMENT
ARRAY       DB    33H, 22H, 99H, 0EEH, 11H, 55H, 44H, 88H
COUNT       EQU  8
D_HERE      ENDS

C_HERE      SEGMENT
            ASSUME CS: C_HERE, DS:D_HERE
START:      MOV  AX, D_HERE
            MOV  DS, AX
            MOV  BX, COUNT

            DEC  BX

DO_2 :      MOV CX, BX

            MOV SI, 0
DO_1        MOV  AL, ARRAY [SI]
            INC  SI
            CMP  AL, ARRAY [SI]
            JA   AHEAD
            XCHG AL, ARRAY [SI]
            MOV  ARRAY [SI-1],AL
AHEAD :     LOOP DO_1
            DEC  BX
            JNZ  DO_2
            INT  3
C_HERE      ENDS
            END  START
```

பயிற்சிகள்

1) 8086 நுண்முறைப்படுத்தியில் எத்தனை தரவு மற்றும் முகவரி கோடுகள் உள்ளன? 8086ஆல் எத்தனை நினைவகங்களை நேரடியாக அணுக முடியும்?

2) செயலூாக்கமுடைய முகவரி (EA) மற்றும் தரவு சார்ந்த முகவரி ஆகியவற்றை விளக்குக.

3) ஊர்தி இடைப்பரிமாற்று அலகின் வேலையை விளக்குக.

4) செயல்பாட்டு அலகைப் பற்றி விளக்குக.

5) கட்டளைச் சுட்டிக்காட்டியின் வேலை என்ன?

6) நிலைப்பாட்டுக் கொடிகள் யாவை? அவை எப்போது அமைவு பெறும் அல்லது மறு அமைவு பெறும்?

7) கீழே கொடுக்கப்பட்ட 8086 முனைகளின் செயல்பாட்டை விளக்குக.

i) $DT/\overline{R}$ ii) $\overline{RD}$ iii) $\overline{WR}$ iv) $M/\overline{IO}$

8) குறியீடு மற்றும் தரவுப் பகுதியை விரிவாக விளக்குக.

9) கட்டளை வரிசை என்றால் என்ன?

10) பின்வரும் கட்டளைகளுக்கான முகவரியிடல் முறைகள் யாவை?

i) MOV BX, CX ii) ADD BX, 1000H iii) MOV AL, SUM[BX] iv) MOV BL [1200H]

11) LEA மற்றும் LDS கட்டளைகளுக்கிடையே உள்ள வேறுபாடுகளைக் கூறுக.

12) தரவு மாற்றுக் கட்டளைகளைப் பற்றி விளக்குக.

13) SUB BX, CX மற்றும் CMP BX, CX ஆகிய இரண்டு கட்டளைகளுக்கிடையேயான வேறுபாடு என்ன?

14) CALL மற்றும் RET கட்டளைகளைத் தகுந்த படத்துடன் விளக்குக.

15) கம்பிக் கட்டளைகளை விளக்குக.

16) LOOP மற்றும் LOOPZ ஆகிய இரண்டு கட்டளைகளுக்கிடையே உள்ள வேறுபாட்டைக் கூறுக.

17) MOV கட்டளைகளுக்கான கட்டளை மாதிரியைப் பயன்படுத்திக் கீழே கொடுக்கப்பட்ட கட்டளைகளுக்கான இயந்திக் குறியீட்டை உருவாக்குக.

i) MOV AX, BX ii) MOV AX, [SI]
iii) MOV BX, SI iv) MOV CS : [BX], AL

18) கூட்டாளர் என்றால் என்ன? அவற்றின் முக்கியத்துவத்தை விளக்குக.

19) SEGMENT, PROC, ENDS, DB, DW ஆகிய கூட்டாளர் வழிமுறைகளை விளக்குக.

20) 8086இன் தடங்கல் சுட்டிக்காட்டி அட்டவணையின் தேவையைப் பற்றி விளக்குக,

21) வகை-2 தடங்கலுக்கு, தடங்கல் சுட்டிக்காட்டி அட்டவணையில், எந்த முகவரி பயன்படுத்தப்படுகிறது?

22) மென்பொருள் தடங்கல்களை விளக்குக.

5

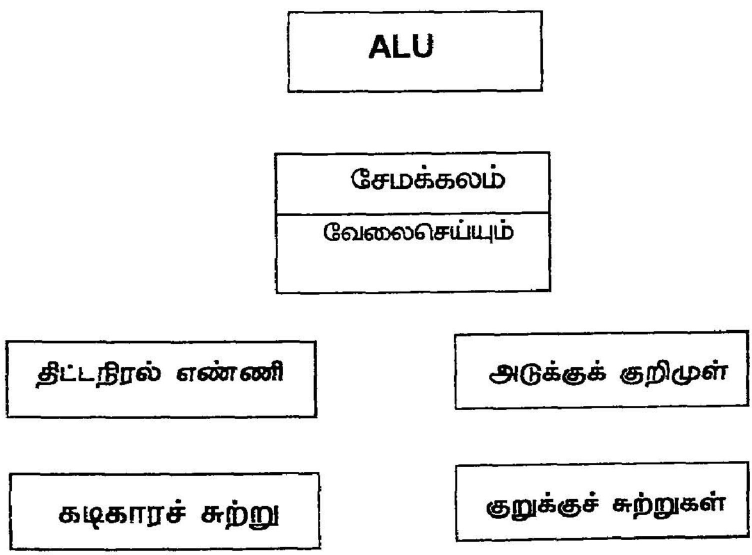
8051 நுண்கட்டுப்படுத்தியின் கட்டுமானம் மற்றும் திட்டநிரல்படுத்தல்

5.1 நுண்முறைப்படுத்திகள் மற்றும் நுண்கட்டுப்படுத்திகள் :

நுண்முறைப்படுத்திகளும் நுண்கட்டுப்படுத்திகளும் ஒரே மாதிரியான அடிப்படைக் கருத்தினைக் கொண்டவை. அவற்றிற்கிடையேயான வேறுபாடுகள்.

நுண்முறைப்படுத்திகள் :

நுண்முறைப்படுத்தி என்பது, ஒரு பொது நோக்குடைய எண்ணிலக்கக் கணிப்பொறி நடுவண் செயலாக்க அலகு (Digital Computer Central Processing Unit) இருப்பினும், வழக்கமான முறையில் இதனை, "ஒரு சில்லில் கணிப்பொறி" என்று அறியலாம். நுண்முறைப்படுத்தி ஒரு முழுமையான எண்ணிலக்கக் கணிப்பொறி என்று அர்த்தமில்லை.



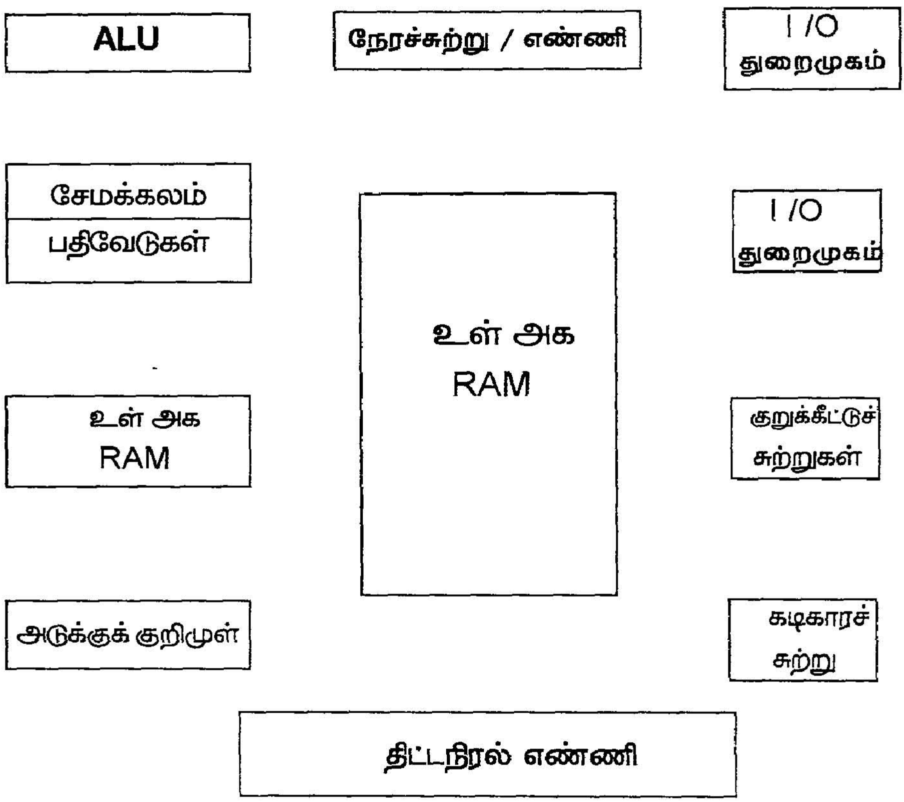
படம் 5.1. நுண்முறைப்படுத்தியின் கட்ட வரைபடம்

படம் 5.1இல் நுண்முறைப்படுத்தி CPUஇன் கட்ட வரைபடம் காட்டப்பட்டுள்ளது. இது எண்கணக்கியல் மற்றும் வாதியல் அலகு (ALU), ஒரு திட்டநிரல் எண்ணி (PC) , ஒரு அடுக்குக் குறிமுள் (SP), சில வேலை செய்யும் பதிவேடுகள், ஒரு கடிகார நேரச்சுற்று மற்றும் குறுக்கீடு சுற்றுகளைக் கொண்டது.

ஒரு முழுமையான நுண்கணிப்பொறியை உருவாக்க, மேலே கூறியவைகளுடன் நினைவகம், பொதுவாகப் படிக்க மட்டும் திட்டநிரல் நினைவகம்

(ROM) மற்றும் வேட்பிடம் அணுகல் தரவு நினைவகம் (RAM), நினைவகக் குறியீட்டுப் பெயர்ப்பி (Memory Decoders), ஒரு அலையியற்றி மற்றும் அதிக உள்ளீடு / வெளியீடு (I/O) கருவிகள் ஆகியவற்றைச் சேர்க்க வேண்டும்.

நுண்முறைப்படுத்தியின் முக்கியமான பயன்பாடு, தரவினைப் படித்து, அத்தரவின் மீது கணக்கீடு செய்து, அதனைச் சேமிப்புக் கருவிகளில் சேமித்து வைப்பதாகும் அல்லது பயனாளிகளுக்குக் கொடுப்பதாகும். நுண்முறைப்படுத் தியினால் பயன்படுத்தப்பட்ட திட்டநிரல்கள், சேமிப்புக் கருவிகளில் சேமிக்கப்படு கின்றன. மேலும் அவை RAMஇல் சுமையேற்றப்படும். சில நுண்முறைப்படுத்தித் திட்டநிரல்கள் ROMயிலும் சேமிக்கப் படுகின்றன.



படம் 5.2. நுண்கட்டுப்படுத்தியின் கட்ட வரைபடம்

ஒரு நுண்கட்டுப்படுத்தியின் கட்ட வரைபடம் படம் 5.2இல் காட்டப்பட்டுள்ளது. இது உண்மையில், ஒரு "சில் மீதான கணிப்பொறி" ஆகும். இது நுண்முறைப்படுத்தியைப் போன்று ALU, PC, SP மற்றும் பதிவேடுகளைக் கொண்டது. இதனுடன் ROM, RAM இணை I/O, தொடர் I/O எண்ணிகள் மற்றும் கடிகாரச் சுற்று ஆகியன இணைக்கப்பட்டுள்ளன.

நுண்முறைப்படுத்தியைப் போன்று, நுண்கட்டுப்படுத்தியும் ஒரு பொது நோக்கான கருவியாகும். ஆனால், தரவினைப் படித்து, சில குறிப்பிட்ட கணக்கீடுகளை மட்டும் தரவின் மீது செயல்படுத்தும். அந்தக் கணக்கீட்டின் அடிப்படையில், சுற்றுப்புறத்தைக் கட்டுப்படுத்தும். நுண்கட்டுப்படுத்தியின் முதன்மைப் பயன் ROM இல் சேமித்து வைக்கப்பட்டுள்ள நிலையான திட்டநிரலைப் பயன்படுத்தி, ஒரு எந்திரத்தின் செயல்பாட்டைக் கட்டுப்படுத்துவதாகும். அது அந்த அமைப்பின் வாழ்நாள் வரையிலும் மாறாது.

நுண்முறைப்படுத்திகளுக்கும் நுண்கட்டுப்படுத்திகளுக்குமிடையே உள்ள ஒற்றுமை : (Comparing Microprocessors and Micro controllers)

நுண்முறைப்படுத்திகளில் தரவினைப் புற நினைவகத்திலிருந்து CPU-க்கு நகர்த்துவதற்காக, அதிக செயற்பாட்டுக் குறியீடுகள் (Opcodes) உள்ளன. நுண்கட்டுப்படுத்திகளில் ஒன்று அல்லது இரண்டு இருக்கக்கூடும். நுண்முறைப்படுத்திகளில் ஒன்று அல்லது இரண்டு வகையான பிட்டுகளைக் கையாளும் கட்டளைகள் இருக்கக்கூடும். நுண்கட்டுப்படுத்திகள் எண்ணற்றவைகளைக் கொண்டிருக்கும்.

சுருங்கக் கூறினால், நுண்முறைப்படுத்தி, குறியீடு மற்றும் புற முகவரிகளிலிருந்து தரவினைச் சில்லுக்கு அதிவேகமாக நகர்த்துவதற்கு அக்கறை கொண்டிருக்கும். நுண்கட்டுப்படுத்தி, சில்லினுள்ளே பிட்டுகளை அதிவேகமாக நகர்த்துவதற்கு அக்கறை கொண்டிருக்கும். புற எண்ணிலக்கப் பகுதிகள் கூடுதலாக இல்லாமல், நுண்கட்டுப்படுத்தி ஒரு கணிப்பொறியாகச் செயல்படும். நுண்முறைப்படுத்தி செயல்படுவதற்குக் கண்டிப்பாக எண்ணற்ற கூடுதல் பகுதிகள் தேவைப்படும்.

வ.எண்.	நுண்முறைப்படுத்தி	நுண்கட்டுப்படுத்தி
1.	நுண்முறைப்படுத்தியில் ALU, பொதுநோக்கப் பதிவேடுகள், அடுக்குக் குறிமுள், திட்டநிரல் எண்ணி, நேரமுறைச் சுற்று மற்றும் குறுக்கீட்டுச் சுற்றுகள் உள்ளன.	நுண்கட்டுப்படுத்தியில் நுண்முறைப்படுத்திச் சுற்றுடன் கடிகார ROM, RAM, I/O கருவிகள், நேரச்சுற்றுகள் மற்றும் எண்ணிகள் உள்ளன.
2.	இது நினைவகம் மற்றும் CPU இடையே தரவினை நகர்த்துவதற்கு நிறைய கட்டளைகளைக் கொண்டிருக்கும்.	இதில் ஒன்று அல்லது இரண்டு கட்டளைகள், தர வினை நினைவகம் மற்றும் CU இடையே நகர்த்துவதற்காக உள்ளன.

வ.எண்.	நுண்முறைப்படுத்தி	நுண்கட்டுப்படுத்தி
3.	இதில் ஒன்று அல்லது இரண்டு வகையான பிட்டுகளைக் கையாளும்கட்டளைகள் உண்டு.	இதில் நிறைய பிட்டுகளைக் கையாளும் கட்டளைகள் உண்டு
4.	நினைவகம் மற்றும் கருவிகளை அடைவதற்கான நேரம் மிக அதிகம்.	I/O கட்டப்பட்டுள்ள நினைவகம் (built-in-memory), மற்றும் I/O கருவிகளை அடைவதற்கான நேரம் மிகக் குறைவு.
5.	நுண்முறைப்படுத்தி போன்ற அமைப்பிற்கு, அதிய வன்மியம் வேண்டியுள்ளது.	நுண்கட்டுப்படுத்தி அமைப்பிற்குக் குறைவான வன்மியம் தேவை. PCB-யின் அளவைக் குறைத்து விடும் மற்றும் நம்பகத்தன்மையை அதிகரிக்கும்.
6.	நுண்முறைப்படுத்தியின் திட்டம் எளிதாக வளையக்கூடியது.	திட்டமிடுதலில் எளிதாக வளையக் கூடியதல்ல.
7.	இதில் ஒரேயொரு நினைவகப் படமம், தரவு மற்றும் குறியீட்டிற்காக உள்ளது.	இதில் தனித்தனியான நினைவகப் படமம், தரவு மற்றும் குறியீட்டிற்காக உள்ளன.
8.	குறைந்த எண்ணிக்கையில், முனைகள் பல செயல்களைச் செய்யக்கூடியன.	அதிக எண்ணிக்கையில், முனைகள் பல செயல்களைச் செய்யக் கூடியன.

5.2 8051 நுண்கட்டுப்படுத்தி வன்பொருள் :

8051 நுண்கட்டுப்படுத்தியின் இனப்பகுதி எண், ஒரு முழுமையான நுண்கட்டுப்படுத்தி குடும்பத்தை உட்கொண்டது. அதனுடைய எண் 8031 முதல் 8751 வரை உண்டு. அவை N வழி உலோக ஆக்ஸைட் சிலிகான் (N - Channel Metal Oride Silicon - NCMOS) மற்றும் நிரப்பு உலோக ஆக்ஸைட் சிலிகான் (Complementary Metal Oride Silicon - CMOS) கட்டுமானம், என்று வேறுபட்ட கட்டு வகைகளில் கிடைக்கும். 8051யின் தனிப்பட்ட சிறப்பு அம்சங்கள்.

- உள்அக ROM மற்றும் RAM
- திட்டமிடத்தக்க முனைகளுடன் I/O துறைமுகங்கள்
- நேரச்சுற்று மற்றும் எண்ணிகள்
- தொடர் தரவு அனுப்புதல்

8051யின் கட்டுமானம், சில குறிப்பிட்ட விசேஷ அம்சங்களால் ஆக்கப்பட்டுள்ளன. அவை

- எட்டு பிட் CPU உடன் பதிவேடு A (சேமக்கலம்) மற்றும் B
- பதினாறு பிட் திட்டநிரல் எண்ணி (PC) மற்றும் தரவுக் குறியீடு (DPTR)
- எட்டு பிட் திட்டநிரல் நிலைப்பாட்டுச் சொல் (PSW)
- எட்டு பிட் அடுக்குக் குறியீடு (SP)
- உட்புற ROM அல்லது EPROM (8751) இல் 0 (8031) முதல் 4K (8051) வரை
- 128 பைட்டுகளுடைய உள்அக RAM :
- நான்கு பதிவேடு வைப்பகங்கள், ஒவ்வொன்றும் எட்டுப் பதிவேடுகளைக் கொண்டவை.
- பதினாறு பைட்டுகள், பிட் மட்டத்தில் கவனிக்கக் கூடியவை.
- எண்பது பைட்டுகளுடைய பொது நோக்குத் தரவு நினைவகம்.
- 32 உள்ளீடு / வெளியீடு முனைகள், நான்கு 8 பிட் துறைமுகங்களாக வைக்கப்பட்டுள்ளன. P0 - P3.
- இரு 16 பிட் நேரச்சுற்று / எண்ணிகள் To மற்றும் T1
- முழு இருமடித் தொடர் தரவு ஏற்பி / பரப்பி SBUF
- கட்டுப்பாட்டுப் பதிவேடுகள் TCON, TMOD. SCON, PCON, IP மற்றும் IE
- இரு புற மற்றும் மூன்று உள்அகக் குறுக்கீட்டு மூலங்கள்
- அலையியற்றி மற்றும் கடிகாரச் சுற்றுகள்.

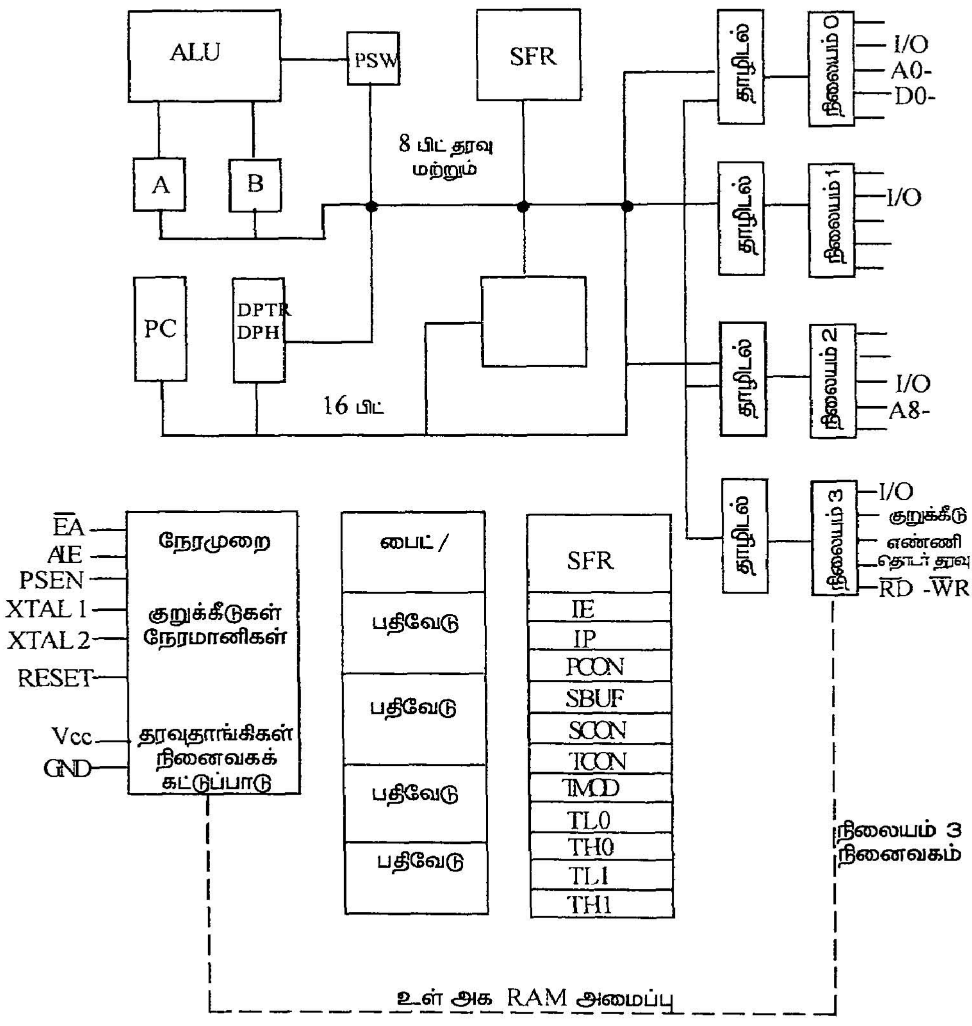
மென்பொருள் கட்டளைகளைப் பயன்படுத்திப் பதிவேடுகள் மற்றும் நினைவக இருப்பிடங்கள் செயல்படும். பெரும்பாலான பதிவேடுகள், ஒரு தனிப்பட்ட வேலையை ஏற்றுக்கொண்டிருக்கும். A அல்லது THO அல்லது PC போன்ற குறிப்பெயருடன் ஒரு தனிப்பட்ட கட்டத்தில் வசிப்பன அவை பொதுவாக, ஒன்றுக்கொன்று பிரித்தறிய முடியாதவை. ஒரு பெரிய கட்டத்தில் தொகுதியாகக் காணப்படும். அப்படிப்பட்டவை உள்அக ROM அல்லது RAM நினைவகம் ஆகும்.

ஒவ்வொரு பதிவேடும் திட்டநிரல் எண்ணியில் மாறுபட்டிருக்கும். அவற்றிற்கு உள்அக 1 பைட் முகவரி நியமிக்கப்பட்டிருக்கும். சில பதிவேடுகள், பிட் மற்றும் பைட்டுகளால் முகவரியிடக் கூடியன.

அதாவது, அந்தப் பதிவேடு முகவரிகளிலுள்ள தரவின் முழு பைட்டும் படிக்கப்படலாம் அல்லது மாற்றப்படலாம் அல்லது தனிப்பட்ட பிட்டுகள் படிக்கப்படலாம் அல்லது மாற்றப்படலாம். மென்பொருள் கட்டளைகள், பொதுவாக ஒரு பதிவேட்டை அதனுடைய முகவரி, குறிபெயர் அல்லது இரண்டும் கொண்டு குறிப்பிடுகின்றன.

5.3 8051-யின் கட்டுமானம் :

8051-யின் உள் அகக்கட்ட வரைபடம் படம் 5.3-இல் காட்டப்பட்டுள்ளது. இருவகையான நினைவகப் பகுதிகள் (தரவு நினைவகம் - RAM மற்றும் திட்டநிரல் நினைவகம் - EPROM / ROM), உள்ளீடு / வெளியீடு துறைமுகங்கள். சிறப்புச்செயல்சார் பதிவேடுகள் மற்றும் மாறுபட்ட புறப்பரப்பு சார்ந்த செயல்களுக்குத் தேவையான கட்டுப்பாட்டு வாதியல் ஆகியவை CPU-இல் உள்ளன. இவை ஒவ்வொன்றும் ஒரு எட்டு பிட் தரவு ஊர்தியின் மூலமாக அனுப்பப்படும். சில்களின் எல்லாப் பகுதிகளிலும் செல்லக்கூடிய உள்அகத்தரவு ஊர்தியை, இது குறிக்கும்.



படம் 5.3. 8051 கட்டுமான வரைபடம்

5.3.1 பதிவேடு வரிசை :

பதிவேடு AC (சேமக்கலம்) :

சேமக்கலம் என்பது, ஒரு 8 பிட் பதிவேடு ஆகும். இதனை ACC அல்லது A பதிவேடு என அழைக்கலாம். இது இயமிகளை வைத்திருக்கும் மற்றும் எண் கணக்கியியல் கட்டளைகளின் (கூட்டல், கழித்தல், பெருக்கல் மற்றும் வகுத்தல்) முடிவினை ஏற்றுக்கொள்ளும். வாதியல் செயல்பாடுகளின்மூலம் சேமக்கலம் இருக்கும் அல்லது சேருமிடமாக, நோக்கீட்டு அணிப்பட்டிகள் (Look up table) மற்றும் புற RAM விரிவாக்கம் போன்ற எண்ணற்ற சிறப்புத் தரவு இயக்கக் கட்டளைகளின் இருப்பிடமாக இது இருக்கக்கூடும். சேமக்கலத்திற்குரிய தனிப்பட்ட செயல்பாடுகளைக் கொண்டுள்ளது. சுழற்று, ஒப்புமை கணக்கிடுதல், சுழற்சிக்கான சோதனை ஆகியவற்றுடன் மேலும் பல செயல்களை செய்யக்கூடியதாகும்.

பதிவேடு : B

B பதிவேடு பெருக்கல் மற்றும் வகுத்தல் போன்ற செயல்களுக்குப் பெரிதும் பயன்படுகிறது. இந்தச் செயல்களில் ஈடுபடாதபோது, ஒரு பொதுநோக்கப் பதிவேடாக செயல்படும். பிற செயல்களுக்கு, இதனை ஒரு கீறலட்டைப் பதிவேடாகப் (Scratch pad Register) பயன்படுத்திக் கொள்ளலாம். இது ஒரு சிறப்புச் செயல்சார் பதிவேடாகும்.

திட்டநிரல் நிலைப்பாட்டுச் சொல் (கொடிப் பதிவேடு) :
(Program statue word (Flag register)) :

இதுவும் ஒரு சிறப்புச் செயல்சார் பதிவேடாகும். இந்த PSW பதிவேடு, திட்டநிரலின் நிலைப்பாட்டைக் கொண்டிருக்கும். இதனைப் பைட் மற்றும் பிட் ஆகிய இரண்டு முறைகளிலும் முகவரியிடலாம். உள்அக RAM-லுள்ள நான்கு பதிவேடு வைப்பகங்களில், நமக்குத் தேவையான பதிவேடு வைப்பகத்தைத் தேர்வு செய்வதற்கு, இது பயன்படுகின்றது. படம் 5.4இல் திட்டநிரல் நிலைப்பாட்டுச் சொல்லின் பிட் மாதிரி காட்டப்பட்டுள்ளது. இது ஒரு 8 பிட் சொல், கீழ்கண்டச் செய்திகளைக் கொண்டிருக்கும்.

B7	B6	B5	B4	B3	B2	B1	B0
CY	AC	F0	RS1	RS0	OV	-	P

CY	பிட் 7	-	சுமைக் கொடி
AC	பிட் 6	-	BCD செயல்பாடுகளுக்கான துணைச் சுமைக் கொடி
F0	பிட் 5	-	பயனாளரின் வரையறுக்கப்பட்ட கொடி (கொடிச் சுழி)
RS1, RS0	பிட் 4-3	-	வேலை செய்யும் பதிவேடு வைப்பகங்களைக் கீழ்க்கண்டவாறு தேர்வு செய்யலாம்.
OV	பிட் 2	-	பொங்கி வழி கொடி
	பிட் 1	-	நிறுத்தி வைக்கப்பட்டது
P	பிட் 0	-	ஒப்புமைக் கொடி (1 = இரட்டைப் படை ஒப்புமை)

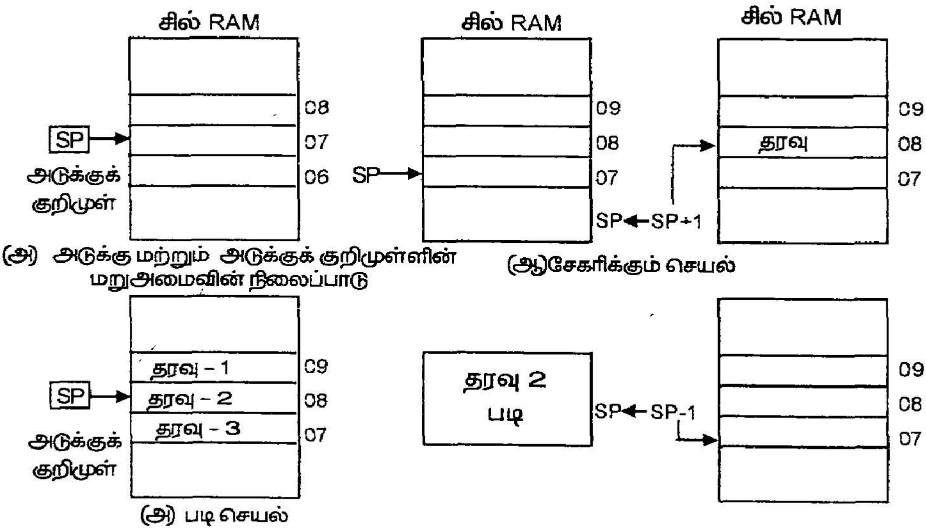
RS1	RS0	வைப்பகம் தேர்வு	
0	0	00H - 07H	வைப்பகம் 0
0	1	08H - 07H	வைப்பகம் 1
1	0	10H - 17H	வைப்பகம் 2
1	1	18H - 1FH	வைப்பகம் 3

படம் 5.4. திட்டநிரல் நிலைப்பாட்டுச் சொல்

அடுக்கு மற்றும் அடுக்குக் குறிமுள் :

அடுக்கு என்பது, உள்அக RAM-இன் ஒரு பகுதியைக் குறிக்கும். இது சில தரவு இயமிகளுடன் இணைந்து, தரவினை வேகமாகச் சேமிக்க அல்லது திரும்பி எடுப்பதற்குப் பயன்படும். 8051யினால் பயன்படுத்தக்கூடிய அடுக்குக் குறிமுள் பதிவேடு உள்அக RAM முகவரியை வைத்திருக்கும். இது அடுக்கின் உச்சி (top of the stack) என்று அழைக்கப்படும். இது 8 பிட் கொண்ட ஒரு சிறப்புச் செயல்சார் பதிவேடாகும். RAM-இன் எந்தப் பகுதியில் வேண்டுமானாலும் அடுக்கு இருக்கலாம்.

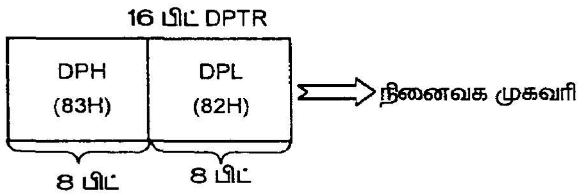
PUSH மற்றும் CALL கட்டளைகளின்பொழுது, தரவு சேமிப்பதற்கு முன்னால் அடுக்குக் குறிமுள்ளின் மதிப்பு அதிகரிக்கப்படும். POP மற்றும் RET கட்டளைகளின் பொழுது, தரவினைத் திருப்பிக் கொடுத்த பின்னர் அடுக்குக் குறிமுள்ளின் மதிப்பு குறைக்கப்படும். மறு அமைவுக்குப் பின், அடுக்குக் குறிமுள் 07F1 என இருக்கும். இதன்மூலம் அதற்கு அடுத்த இருப்பிடமான 08H-இல் முதல் தரவைச் சேமித்துக்கொள்ளலாம். அடுக்கு மற்றும் அடுக்குக் குறிமுள்ளின் செயல்பாடுகள், படம் 5.5-இல் உதாரணங்களால் தெளிவாகக் காட்டப்பட்டுள்ளன.



படம் 5.5 அடுக்கின் செயல்பாடு

தரவுக் குறிமுள் : (Data Pointer)

தரவுக் குறிமுள் (DPTR), உயர் பைட் (DPH) மற்றும் தாழ் பைட் (DPL) ஆகிய இரண்டு 8 பிட் பதிவேடுகளாக இருக்கும். இதன் செயல் 16 பிட் முகவரியை வைத்திருப்பதாகும். இதனை 16 பிட் கொண்ட ஒரே பதிவேடாகவும் அல்லது 8 பிட் கொண்ட இரண்டு பதிவேடுகளாகவும் பயன்படுத்திக் கொள்ளலாம்.



மறைமுக தாவல், நோக்கீட்டு பட்டி கட்டளைகள் மற்றும் புற தரவினை மாற்றல் போன்றவற்றிற்கு அடி பதிவேடாக ஊழியம் செய்யும். DPTR ஒரு தனிப்பட்ட உள்அக முகவரியினைப் பெற்றிருக்காது. DPH (83H) மற்றும் DPL (82H) வெவ்வேறான உள்அக முகவரிகளைக் கொண்டிருக்கும்.

திட்டநிரல் எண்ணி :

8051 நுண்கட்டுப்படுத்தியிலுள்ள திட்டநிரல் எண்ணி, 16 பிட்டுகளைக் கொண்டது. திட்டநிரல் எண்ணி எந்த முகவரியை வைத்துள்ளதோ, அந்த நினைவக இருப்பிடங்களிலுள்ள திட்டநிரல் கட்டளைக்கான பைட்டை, நுண்கட்டுப்படுத்தி அடுத்துச் செயல்படுத்தும். அதாவது, அடுத்ததாக எந்த நினைவக இருப்பிடங்களிலுள்ள கட்டளை பைட்டைச் செயல்படுத்த வேண்டுமோ, அதற்குரிய முகவரி PC-இல் இருக்கும். PC-இன் மதிப்பானது, கட்டளை செயல்படுகின்றபோது அதன் தன்மையைப் பொறுத்துத் தானாகவே ஒன்று அதிகரிக்கின்றது அல்லது மாறுபடுகின்றது. PC பதிவேட்டிற்கு மட்டுமே, எவ்வித உள்அக முகவரியும் கிடையாது.

தொடர் தரவுத் தாங்கி : *(Serial data buffer)*

இது கடத்துத் தாங்கிப் பதிவேடு (transmit buffer register) மற்றும் திரட்டுத் தாங்கிப் பதிவேடு (receive buffer register) என இரண்டு தனித்தனி பதிவேடுகளைக் கொண்டிருக்கும். கடத்துத் தாங்கிப் பதிவேடு, இணை – உள்ளே – தொடர் – வெளி (parellel in - serial out) முறையில் செயலாற்றுகின்ற பதிவேடாகும். திரட்டுத் தாங்கி பதிவேடு, தொடர் – உள் – இணை – வெளி (serial - in - parallel out) முறையில் செயல்படுகின்ற ஒரு பதிவேடாகும். SBUF-க்குத் தரவு செல்கின்றபோது, அது கடத்தித் தாங்கிக்குச் சென்று தொடர் அனுப்புதலை நடைமுறைப்படுத்துகிறது. SBUF-லிருந்து தரவு வருவதாக இருந்தால், அது திரட்டுத் தாங்கிப் பதிவேட்டிலிருந்து வருகின்றது.

நேரச்சுற்றுப் பதிவேடுகள் : (Timer registers)

TH0, TL0 மற்றும் TH1, TL1 என்கிற இணையான பதிவேடுகள், 16 பிட் கொண்ட எண்ணிப் பதிவேடுகளாகும். இவைகள் நேரச்சுற்று / எண்ணிகள் 0 மற்றும் 1 ஆகிய செயல்களுக்குப் பயன்படுகின்றன.

கட்டுப்பாடு பதிவேடுகள் : (Control registers)

இது IP. IE, TMOD, TCON, SCON மற்றும் PCON என்கிற சிறப்புச் செயல்சார் பதிவேடுகள், குறுக்கீடுகள் , நேரச்சுற்று / எண்ணிகள் மற்றும் தொடர் துறைமுகம் ஆகியவற்றிற்குரிய கட்டுப்பாடு மற்றும் நிலைப்பாட்டுத் தகவல்களைக் கொண்டிருக்கும்.

5.3.2. நேரமுறை மற்றும் கட்டுப்பாடு அலகு : (Timing and control unit)

இந்த அலகானது, சுற்றின் உட்பகுதியில் நடைபெறுகின்ற செயல்களுக்குத் தேவையான அனைத்து நேரமுறை மற்றும் கட்டுப்பாட்டுச் சைகைகளை ஏற்படுத்துகின்றது. மேலும் இது, வெளிப்புறத்திலுள்ள அமைப்பு ஊர்தி (system bus)களைக் கட்டுப்படுத்தத் தேவையான கட்டுப்பாட்டுச் சைகைகளையும் உண்டாக்குகிறது.

5.3.3. அலையியற்றி : (oscillator)

இது படிக அலையியற்றியைப் பயன்படுத்தி, நுண்கட்டுப்படுத்தி செயல்படுவதற்குத் தேவையான அடிப்படை நேரமுறை கடிகாரச் சைகைகளை உருவாக்குகின்றது.

கட்டளைப் பதிவேடு : (Instruction Register)

இந்தப் பதிவேடு, நுண்கட்டுப்படுத்தி செய்து முடிக்க வேண்டியவற்றைச் செய்யக் கட்டளைகளின் செயற்பாட்டுக் குறியீடுகளை (opcode) குறியீட்டுப் பெயர்ப்பி செய்கின்றது. மேலும் நேரமுறை மற்றும் கட்டுப்பாட்டு அலகுக்குத் தகவலைக் கொடுத்துக் கட்டளைகளைச் செய்துமுடிக்கத் தேவையான சைகைகளை உருவாக்குகிறது.

EPROM **மற்றும் திட்டநிரல் முகவரிப் பதிவேடு :**

IC சில்லினுள் EPROM-இல் இந்தப் பதிவேடுகள் அமையப்பெற்றிருக்கும். இவைகள் உட்புறமாக முகவரியிடும் தன்மையைப் பெற்றிருக்கும்.

RAM and RAM **முகவரிப் பதிவேடு :**

இத்தகைய பதிவேடுகள் உட்புறமாக அமைந்துள்ள RAM-இன் 128 பைட்டுகளில் இருக்கும். இவைகளும் உட்புறமாக முகவரியிடும் தன்மையைப் பெற்றிருக்கும்.

ALU :

இந்த அலகு, TMP1 மற்றும் TMP2 ஆகிய தற்காலிகப் பதிவேடுகள் வைத்துள்ள இயமிகளின்மீது, எண் கணக்கியல் மற்றும் வாதியல் செயல்களை 8 பிட்டில் செய்கின்றது. நுண்கட்டுப்படுத்தியின் பயனாளிகள், தற்காலிகப் பதிவேட்டை அடைய இயலாது. அதாவது, இதனைத் திட்டநிரலில் பயன்படுத்த முடியாது.

5.3.4. SFR பதிவேடு வைப்பகம் : (SFR Register Bank)

இது சிறப்புச் செயல்சார் பதிவேடுகள் கொண்ட அமைப்பினைக் குறிப்பிடுகின்றது. அவற்றிற்குரிய முகவரிகளைக் கொடுத்து, இதிலுள்ள பதிவேடுகளை அடைய வழி செய்து கொள்ளலாம். இதிலுள்ள பதிவேடுகளின் முகவரி 80H முதல் FFH வரை இருக்கும்.

குறுக்கீடு, தொடர் துறைமுகம் மற்றும் நேரச்சுற்றுகளுக்குரிய சுற்றுகள், நேரமுறை மற்றும் கட்டுப்பாடு அலகின்மூலம் ஏற்படுத்தப்படுகின்ற கட்டுப்பாட்டுச் சைகைகளைக் கொண்டு அதற்குரிய செயல்களைச் செய்கின்றது.

5.4 8051இன் முனைகள் :

8051 DIP முறையில் உருவாக்கப்பட்ட ஒரு சில்லாகும். இது 40 முனைகளைக் கொண்டுள்ளது. படம் 5.6 இல் 8051-இன் முனை வரைபடம் காட்டப்பட்டுள்ளது. இதில் முக்கியமானது 8051-இன் பல முனைகள், ஒன்றிற்கு மேற்பட்ட வேலைகளைச் செய்யக்கூடியன. முனைகளின் பதிலிச் செயல்கள், தடிம எழுத்துக்களால் காட்டப்பட்டுள்ளன.

5.4.1 நிலையங்கள் :

8051-இல் 32 I/O முனைகள் நான்கு எட்டு - பிட் இணை நிலையங்களாக (P0, P1, P2 மற்றும் P3) வடிவமைக்கப்பட்டுள்ளன. அனைத்து நிலையங்களும் இருதிசையிலானவை. அதாவது, ஒவ்வொரு முனையும் உள்ளீடு அல்லது வெளியீடாக (அல்லது இரண்டும்) வடிவமைக்கப்பட்டிருக்கும். நிலையம் 1-இன் முனைகளைத் தவிர மற்ற நிலையங்களின் முனைகள், பலவின் ஒன்றாக இருக்கும். ஒவ்வொரு நிலையமும் ஒரு தாழிடல், ஒரு வெளியீடு ஓட்டுநர் மற்றும் ஒரு உள்ளீடு தாங்கியைக் கொண்டிருக்கும்.

நிலையம் 0 (முனைகள் 32 - 39) :

நிலையம் 0-இன் முனைகள் I/O முனைகளாகப் பயன்படுத்தப்படும். நிலையம் 0-இன் வெளியீடு தூண்டுதல் மற்றும் உள்ளீடு தாங்கிகள், புற நினைவகத்தை அடைய பயன்படுகின்றன. நிலையம் 0, புற நினைவக முகவரியின் தாழ் வரிசை பைட்டை எழுதுவதற்காகவா அல்லது படிப்பதற்காகவா என நேரப் பல்முனையமைப்பின் தரவின்மூலம் வெளியீடு செய்யும். ஆதலால், நிலையம் 0-வை பல்முனை ஒன்றாக்கி முகவரி / தரவு ஊர்தியாகப் பயன்படுத்தலாம்.

நிலையம் 1 (முனைகள் 1-8) :

நிலையம் 1, இருதிசையில் செயல்படக்கூடிய ஒரு 8 பிட் நிலையமாகும். இது பிட் மற்றும் பைட் முறைகளில் முகவரியிடும் தன்மை கொண்டது. நிலையம் 1-இன் முனைகள், முனைகளாக மட்டுமே பயன்படுத்தலாம்.

நிலையம் 2 (முனைகள் 21 - 28) :

நிலையம் 2-இன் வெளியீட்டுத் தூண்டுதல், புற நினைவகத்தை அடையப் பயன்படும். புற நினைவக முகவரி 16 பிட் அகலமாக இருந்தால், உயர் வரிசை பைட்டை நிலையம் 2-இல் வெளியிடும். இல்லையெனில், நிலையம் 2 I/O நிலையமாகப் பயன்படும். நிலையம் 2-இம் இருதிசையில் செயல்படக் கூடிய ஒரு 8 பிட் I/O நிலையமாகும். இதுவும் பிட் மற்றும் பைட் முறைகளில் முகவரியிடும் தன்மை கொண்டது. இதற்கும் SFR-க்கான முகவரி நெடுக்கத்தில் ஒரு தனி முகவரி உள்ளது.

நிலையம் 3 (முனைகள் 10 - 17)

நிலையம் 3-இல் உள்ள அனைத்து முனைகளும், பல செயல்களைச் செய்யக்கூடியன. தனிப்பட்ட செயல்களுடன் சேர்த்து இரண்டு புறத் தடங்கல்கள், இரண்டு எண்ணி உள்ளீடுகள், இரண்டு குறிப்பிட்ட தரவு வரிகள் மற்றும் இரண்டு நேரக் கட்டுப்பாட்டுச் சைகைகள் கீழே கொடுக்கப்பட்டுள்ளன.

குறி	இடம்	பெயர் மற்றும் பொருள்
$\overline{RD}$	P3.7	வெளியீடு கட்டுப்பாட்டுத் தரவைப் படி. புற நினைவகத் தரவினைப் படிக்கும்போது, வன்மியம் குறைவியக்கக் துடிப்பை உண்டாக்கும்.
$\overline{WR}$	P3.6	வெளியீடு கட்டுப்பாட்டுத் தரவை எழுது. புற நினைவகத் தரவினை எழுதும்போது, வன்மியம் குறைவியக்கத் துடிப்பினை உண்டாக்கும்.
T1	P3.5	நேரமானி / எண்ணி 1 புற உள்ளீடு அல்லது சோதனை முனை.
T0	P3.4	நேரமானி / எண்ணி 0 புற உள்ளீடு அல்லது சோதனை முனை.
$\overline{INT1}$	P3.3	தடங்கல் 1 உள்ளீட்டு முனை. தாழ் மட்டம் அல்லது இறங்கு விளிம்புத் துவக்கி.
$\overline{INT0}$	P3.2	தடங்கல் 0 உள்ளீட்டு முனை. தாழ் மட்ட அல்லது இறங்கு - விளிம்புத் துவக்கி.
TXD	P3.1	UART வகையில் தொடர் துறைமுகத்திற்காகத் தரவு முனையை அனுப்பு. பெயரும் பதிவேடு வகையில் கடிகார வெளியீடு.
RXD	P3.0	UARTவகையில், தொடர் துறைமுகத்திற்காகத் தரவு முனையைப் பெறு. பெயரும்பதிவேடு வகையில்தரவு/முனை.

திறன் வழங்கி முனைகள் Vcc (முனை 40) மற்றும் Vss (முனை 20)

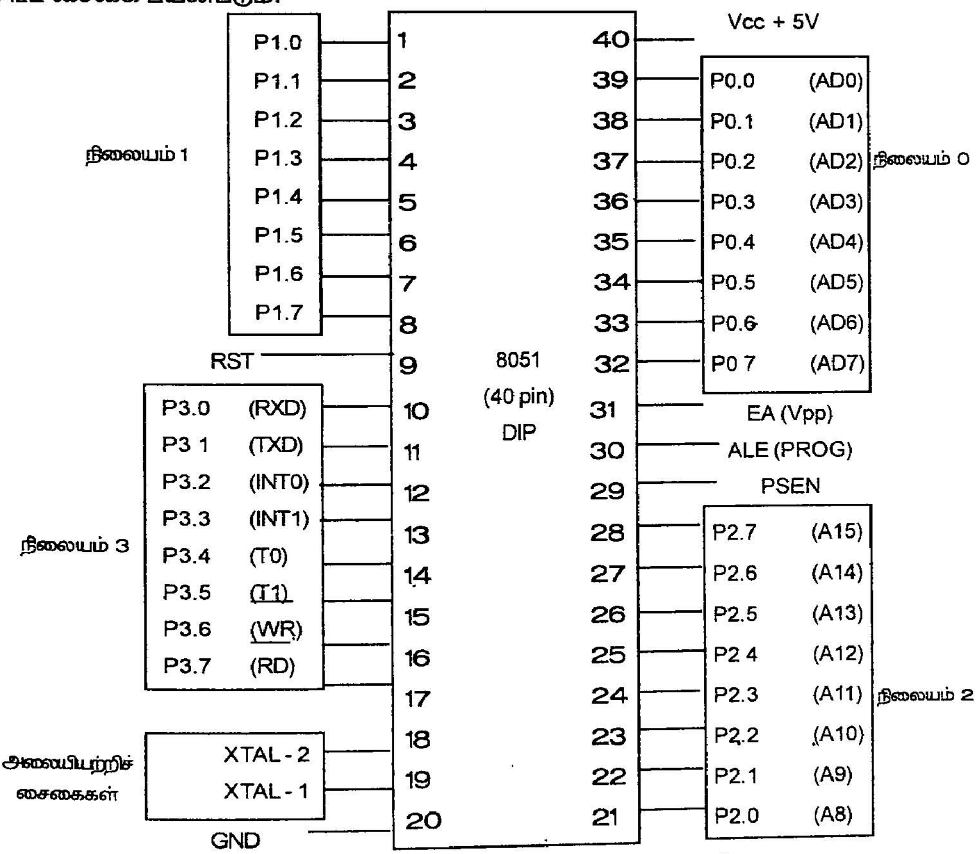
8051 d.c. திறன் வழங்கி + 5V-இல் செயல்படும். முனை Vcc உடன் +5V-ஐ இணைக்க வேண்டும் மற்றும் தரையிடப்பட வேண்டிய முனை 0-இல் இணைக்க வேண்டும்.

அலையியற்றி முனைகள் XTAL2 (முனை 18) மற்றும் XTAL1 (முனை 19).

அகக் கடிகாரச் சைகையினை உண்டாக்குவதற்காக, இந்த இரண்டு முனைகளிலும் புற அலையியற்றி வேண்டும். XTAL1 மற்றும் XTAL2 : நுண்கட்டுப்படுத்தி செயலாற்றும் பொருட்டு, அதற்குத் தேவையான கடிகார அதிர்வெண்ணை உருவாக்குகின்ற வகையில், IC சில்லினுள் ஒரு அலையியற்றி இருக்கும். XTAL1 தலைகீழாக்கி XTAL2 அலையியற்றிப் பெருக்கியின் உள்ளீடாகச் செயல்படும். தலைகீழாக்கி அலையியற்றிப் பெருக்கியின் வெளியீடாகச் செயல்படும்.

ALE (முகவரித் தாழிடல் செயலியக்க முனை 30) (Address latch Enable)

AD_0 - AD_7 வரிகள் பல்வின் ஒன்றாக்கி முனைகளாகும். இந்த வரிகளை ஒன்றின் பலவாக்க மற்றும் முகவரியின் அடிப்பாதியைப் பெறுவதற்காக, 8051இன் புறத் தாழிடல் ALE சைகை பயன்படும்.



படம் 5.7 8051-இன் முனைகள்

RST(மறு அமைவு, முனை 9) : (Reset, Pin 9)

இந்த முனை 8051ஐ மறு அமைவு செய்யப் பயன்படும். சரியான மறு அமைவுச் செயற்பாட்டிற்காக, மறு அமைவுச் சைகையை அலையியற்றி சென்று கொண்டிருக்கும்போதே, இரண்டு எந்திர சுழற்சிகளுக்காவது மறுஅமைவு சைகையை உயர்நிலையில் கண்டிப்பாக வைத்திருக்க வேண்டும்.

$\overline{\text{PSEN}}$ **(திட்டநிரல் சேமிப்பு செயலியக்கம், முனை 29) :** (Program Store Enable, pin 29)

இது குறைவியக்க வெளியீட்டுக் கட்டுப்பாட்டு சைகை, புற ROM / EPROM-இன் செயலியக்கச் சைகையை ஊக்குவிக்க இது பயன்படும். புற நினைவகத்தைப் படிக்கும்போது, ஒவ்வொரு ஆறு அலையியற்றிக் கால அளவிற்கு, இது ஊக்குவிக்கப்படுகின்றது.

$\overline{\text{EA}}$ **(புற வழி அடைதல், முனை 31)** (External Acess, pin 31)

இது புறத் திட்டநிரல் நினைவகத்தை அடைவதற்குரிய முனை ஆகும். புறத் திட்டநிரல் நினைவகத்தை அடையும்பொழுது தாழ் நிலையிலும், மற்றும் அக நினைவகத்திலுள்ள திட்டநிரலைச் செயல்படுத்தும்போது உயர்நிலைக்கும் இது செல்லும். IC-யினுள் அமைந்துள்ள EPROM- ஐ திட்டநிரல் செய்கின்றபோது, இந்த முனைக்கு 21 V DC மின்னழுத்தம் கொடுக்க வேண்டும்.

5.5 நடுவண் செயலாக்க அலகு : (Central Processing Unit) : CPU

8051-இன் CPU எட்டு - பிட் எண் கணிதம் மற்றும் வாதியல் அலகு (Arithmatic and Logic Unit) A,B, PSW மற்றும் SP பதிவேடுகள், 16 பிட் திட்டநிரல் எண்ணி மற்றும் தரவுக் குறிமுள் (Data Pointer) பதிவேடுகள் ஆகியவற்றைக் கொண்டிருக்கும். இந்தப் பதிவேடுகளுடன் சிறப்பு செயல்சார் பதிவேடுகள் குழுவும் உண்டு.

5.6 எண் கணக்கியல் மற்றும் வாதியல் அலகு : (ALU)

நுண்கட்டுப்படுத்தி 8051லுள்ள ALU எண் கணக்கியல் மற்றும் வாதியல் செயல்கள், 8 பிட் மாறிகளில் செய்யப்படுகின்றன. எண் கணக்கியல் அலகு கூட்டல், கழித்தல், பெருக்கல் மற்றும் வகுத்தல் செயல்களை நிறைவேற்றும். வாதியல் அலகு AND, OR, XOR மற்றும் தரவினைச் சுழற்ற, காலி செய்ய மற்றும் நிரப்ப ஆகிய வாதியல் செயல்களைச் செய்யும். 8051 கட்டுமானத்தின் முக்கியமான சிறப்பம்சம் ALU ஆனது ஒரு பிட் மற்றும் எட்டு பிட் தரவு வகைகளைத் திறமையாகக் கையாளும். வாதியல் கணக்கீட்டில் தனிப்பட்ட பிட்டுகளின் அமைவு, காலி, நிரப்பு, நகர், சோதனை செய் போன்ற செயல்களைச் செய்யப் பயன்படும். ALU தேவையான இயமிகள் TMP1 மற்றும் TMP2 என்கிற தற்காலிகப் பதிவேடுகளிலிருக்கும். இந்தத் தற்காலிகப் பதிவேடுகளை நாம் நேரிடையாகக் கட்டளைகளில் பயன்படுத்த முடியாது. இதில் கூட்டல் மற்றும் கழித்தல் செயல்களை மட்டும் குறியீடு (Signed) மற்றும் குறியிடாத (unsigned) ஆகிய இரண்டு முறைகளிலும் செய்யலாம். மற்ற செயல்களைக் குறியிடாத முறையில் மட்டுமே செய்ய முடியும்.

எண் கணக்கியல் மற்றும் வாதியல் செயல்கள் முடிந்த பின்னர் கிடைக்கப்பெறுகின்ற முடிவு, சேருமிடம் பதிவேட்டில் (destination register) சேமிக்கப்படும்.

5.7 தரவு நினைவகம் மற்றும் பதிவேடு வைப்பகம் (Data memory and Register Bank) :

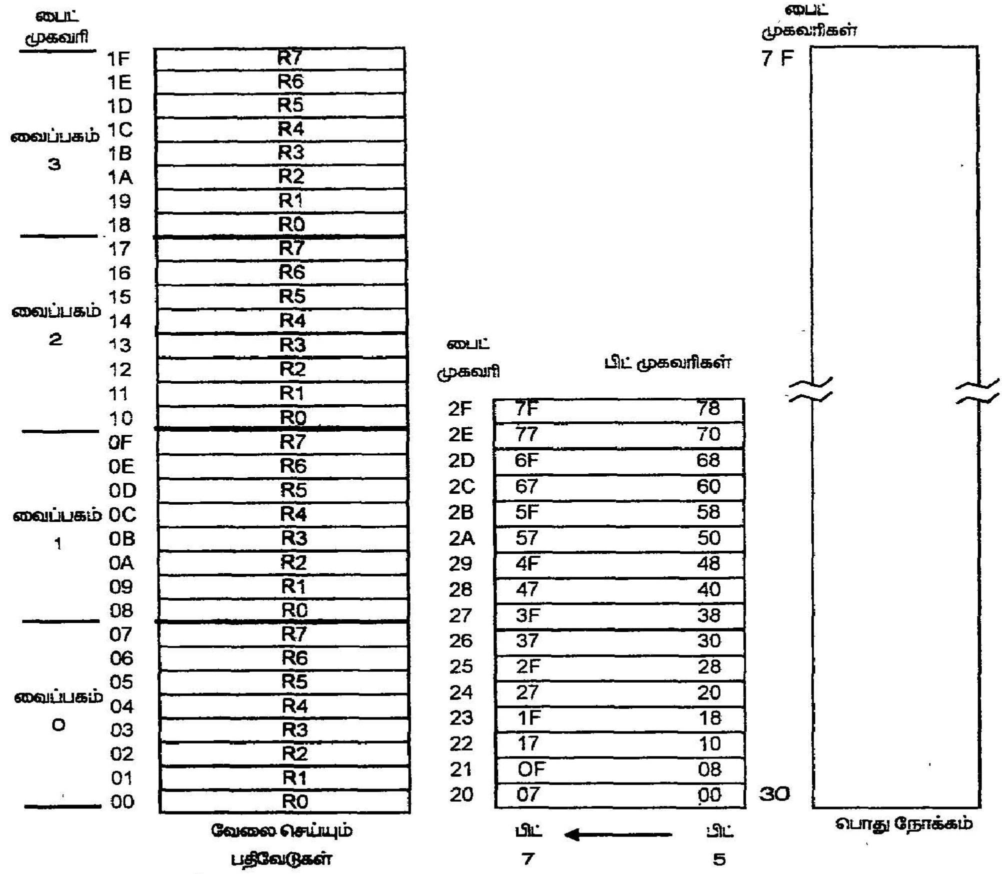
8051, 128 பிட் அக RAM-ஐப் பெற்றிருக்கும். RAM முகவரி பதிவேட்டின்மூலம் இதனை அடையலாம். அக RAM-இன் ஒழுங்கமைப்பினைப் படம் 5.7- இல் காணலாம். அக RAM, மூன்று தெளிவான பகுதிகளைக் கொண்டது.

- வேலை செய்யும் பதிவேடுகள்
- பிட் முகவரியிடக் கூடிய பதிவேடுகள்
- பொதுநோக்கப் பதிவேடுகள்

1. அக RAM-இன் முகவரி 00H முதல் 1FH வரையுள்ள முதல் 32 பைட்டுகள், 32 வேலை செய்யும் பதிவேடுகளாகும். நான்கு வைப்பகமாக, ஒழுங்காக இவை அமைக்கப்பட்டுள்ளன. ஒவ்வொரு வைப்பகத்திலும் எட்டுப் பதிவேடுகள் உள்ளன. 0 முதல் 3 வரை எண்ணால் குறிக்கப்பட்ட நான்கு வைப்பகங்களில், எட்டுப் பதிவேடுகள் R0 - R7 உண்டு. ஒவ்வொரு பதிவேடும், அதனுடைய RAM முகவரி அல்லது பெயரின் மூலம் முகவரியிடலாம். ஒரு நேரத்தில் ஒரே ஒரு வைப்பகம் மட்டும்தான் பயன்பாட்டில் இருக்கும். PSN-லுள்ள RS0 மற்றும் RS1 பிட்டுகள் எந்த வைப்பகம் தற்போது பயன்பாட்டில் உள்ளது என்பதை முடிவு செய்யும். தேர்வு செய்யப்படாத பதிவேடு வைப்பகங்கள், பொதுநோக்க RAM ஆகப் பயன்படும். மறு அமைவின்போது வைப்பகம் 0 தேர்வு செய்யப்படும்.

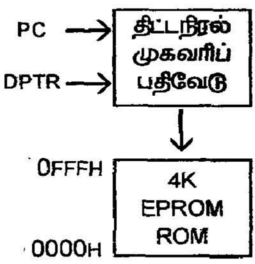
2. 8051, 16 பைட்டுகள் பிட் முகவரியிடக்கூடிய பரப்பைக் கொடுக்கிறது. இது RAM பைட்டின் முகவரிகள் 20H முதல் 2FH வரை ஆக்கிரமித்துக்கொள்ளும். ஆக, மொத்தம் 128 முகவரியிடக்கூடிய பிட்டுகளைப் பெற்றிருக்கும். முகவரியிடக்கூடிய பிட், அதனுடைய முகவரி 00H முதல் 7FH வரை குறிக்கப்படலாம் அல்லது 8 பிட்டுகள் முகவரி 20H முதல் 2FH வரை உருவாகலாம். உதாரணமாக, பிட் முகவரி 4FH என்பது பைட் முகவரி 29இன் பிட்டு 6ஐ குறிக்கும்.

3. RAM பரப்பிற்குமேல் முகவரி 30H முதல் 7FH வரை பிட் முகவரியிடக்கூடிய பரப்பு பொது – நோக்க RAM என அழைக்கப்படும். இதனைப் பைட்டின் மூலமாக முகவரியிடலாம்.



படம் 5.7 8051–இன் அக RAM–இன் ஒழுங்கமைப்பு

5.8 திட்டநிரல் நினைவகம் :



8051–இல் 4 கி பைட் அக RAM-ஐ கொண்டுள்ளது. அதன் முகவரி இருப்பிடம் 0000H முதல் 0FFFH வரை இருக்கும். இது சில்லை அமைக்கும்போது, உற்பத்தியாளரால் திட்டநிரல் செய்யப்பட்டதாகும். அதனைக் கட்டமைத்த பின்னர் அந்தப் பகுதியை அழிக்கவோ அல்லது மாற்றவோ முடியாது. திட்டநிரலின் இறுதிப் படிவத்தை இதில் சேமிக்கப் பயன்படுகிறது.

திட்டநிரல் முகவரிப் பதிவேட்டினைப் பயன்படுத்தித் திட்டநிரல் நினைவகத்தை அடையலாம். திட்டநிரலின் முகவரிகள் 0FFFH-க்கு அதிகமானால், (அக RAM-ன் கொள்திறனைவிட அதிகரித்தால்) 8051 தன்னிச்சையாகப் புறத் திட்டநிரல் நினைவகத்திலிருந்து குறியீடு பைட்டுகளை எடுத்துக்கொள்ளக் காரணமாகும். எதுவாகிலும், குறியீடு பைட்டுகள் புற நினைவக முகவரிகள் 0000H முதல் FFFFH வரை தனிப்பட்ட முறையில், புற அடைமுனையைத் (EA) (External access pin) தரையுடன் இணைத்து எடுத்துக்கொள்ளும்.

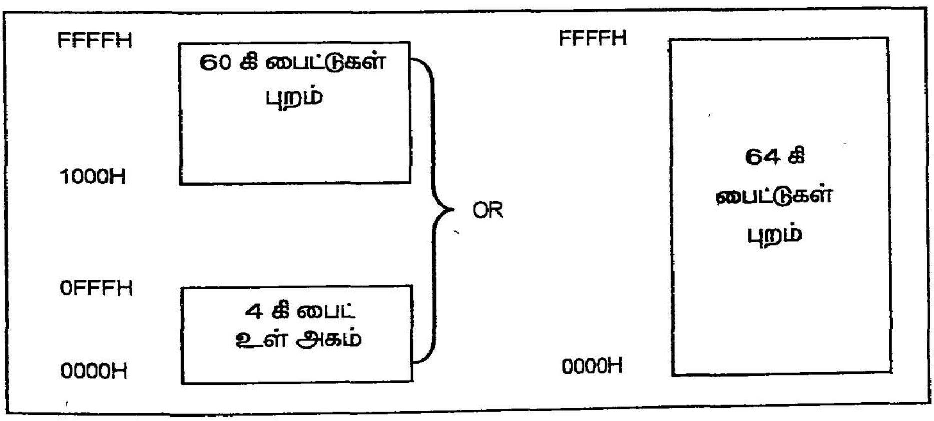
5.9. 8051இன் நினைவக அமைப்பு : (Memory Organisation in 8051)

8051 நுண்கட்டுப்படுத்தி மூலம், 64K பைட்டுகள், திட்டநிரல் நினைவகத்தையும் மற்றும் 64K தரவு நினைகத்தையும் அடையலாம். இதில் 4 கி பைட்டுகள், உள்அகத் திட்டநிரல் மற்றும் 256 பைட்டுகள் உள்அக தரவு நினைவகங்கள் உள்ளன.

ஒவ்வொரு நினைவக வகையும் வேறுபட்ட முகவரியிடும் தன்மை, மாறுபட்ட கட்டுப்பாட்டுச் சைகைகள் மற்றும் வேறுபட்ட செயல்களையும் கொண்டிருக்கும். ஒவ்வொன்றும் சுதந்திரமாக இணைக்கப்படலாம். மேலும், ஒவ்வொன்றும் ஒரே மாதிரியான முகவரி மற்றும் தரவு ஊர்திகளையும், வேறுபட்ட கட்டுப்பாட்டுச் சைகைகளையும் பயன்படுத்தும்.

பொதுவாக, ROM நினைவகம், திட்டநிரல் நினைவகமாகவும் RAM தரவு நினைவகமாகவும் பயன்படுத்தப்படுகின்றன.

புறத் திட்டநிரல் நினைவகம் :



படம் 5.9. 8051 திட்ட நிரல் நினைவகம்

திட்டநிரல் நினைவகம், 64 கி பைட்டுகள் இருப்பிடங்களைப் பெற்றிருக்கும். 64 கி பைட் திட்டநிரல் நினைவகம், உள்அக மற்றும் புற நினைவக இருப்பிடங்களாக அமையப்பட்டிருக்கலாம். படம் 5.9இல் 8051இன் திட்டநிரல் நினைவகத்தின் படம் காட்டப்பட்டுள்ளது.

ROM :

திட்டநிரல் நினைவகம் ROM மற்றும் EPROM கொண்ட சில்களில் அடிப்பக்கம் உள்ள 4 கி பைட்டுகள் சில்களினுள் அமையப் பெற்றிருக்கும். ROM இல்லாத சில்களில் அனைத்துத் திட்டநிரல் நினைவகத்தையும் வெளியில் வைத்துக்கொள்ளலாம். உள்அக மற்றும் புறமாக அமைந்துள்ள திட்டநிரல் நினைவக இருப்பிடங்களை வேறுபடுத்தித் தேர்வு செய்வதற்கு நுண்கட்டுப்படுத்தியிலுள்ள $\overline{EA}$ என்கிற முனை பயன்படுகிறது. $\overline{EA}$=1 எனில், 0000H முதல் 0FFFH வரையுள்ள 4 கி பைட் அளவு கொண்ட முகவரி உள்அக ROM-லிருந்தும், 1000H முதல் FFFFH வரையுள்ள 60கி பைட் அளவு கொண்ட முகவரி புற ROM- லிருந்தும் தாழிடப்படும்.

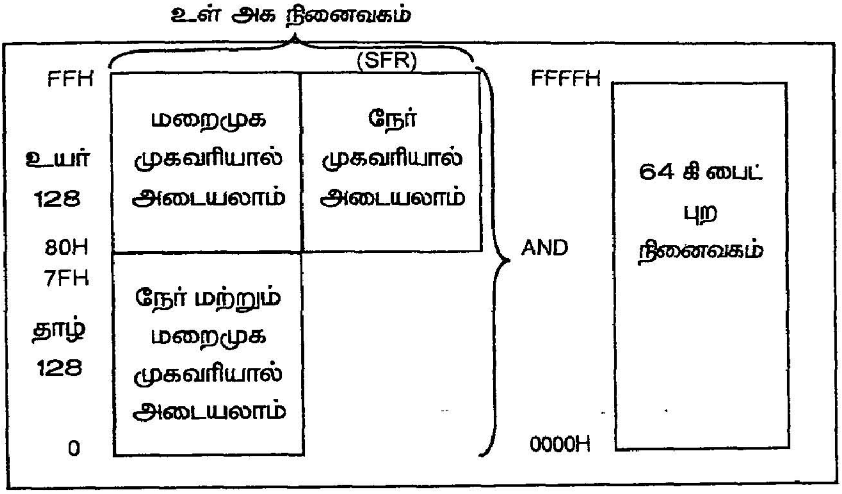
$\overline{EA}$=0 எனில், அனைத்துக் கட்டளைகளும் புறத் திட்டநிரல் நினைவகத்திலிருந்து தாழிடப்படும். $\overline{PSEN}$ சைகை புறத் திட்டநிரல் நினைவகத்தின் படி ஸ்ட்ரோப் சைகையாகச் செயல்படுகின்றது. உள்அகத் திட்டநிரல் நினைவகத்தை அடைய $\overline{PSEN}$ பயன்படுத்தப் படுவதில்லை. திட்டநிரல் நினைவகத்திலுள்ள 00 முதல் 23H வரையுள்ள இருப்பிடங்கள் குறுக்கீடு சேவை செயல்முறைகளுக்குப் (interrupt service routine) பயன்படுகிறது.

புறநினைவகத்தை அடையத் துறைமுகம் 0 மற்றும் துறைமுகம் 2 ஆகிய இரண்டு துறைமுகங்களும் பயன்படுத்தப்படுகின்றன. துறைமுகம் 0 பல்வின் ஒன்றாக்கி முகவரி /தரவு ஊர்தியாகச் செயல்படும். T - சுழற்சியின் தொடக்கத் தாழ் வரிசை 8 பிட் முகவரியை இது கொடுக்கும். இதேபோல், துறைமுகம் 2 ஒரு முகவரி ஊர்தியாகச் செயல்படும். இது உயர்வரிசை 8 பிட் முகவரியைக் கொடுக்கும். தரவு அல்லது திட்டநிரல், துறைமுகம் 0 வழியாகச் செல்லும் அல்லது வரும்.

8051ஆல் உருவாக்கப்பட்ட ALE சைகை மற்றும் புறத்தாழிடலைப் பயன்படுத்தி 8 பிட் முகவரியானது தாழிடப்படும்.

$\overline{EA}$ சைகை புறத் திட்டநிரல் நினைவகத்தைத் தேர்வு செய்வதற்குப் பயன்படுகிறது. $\overline{EA}$=0 எனில், (Vss உடன் இணைக்கப்பட்டிருந்தால்) அனைத்துத் திட்டநிரல்களும் புறத் திட்டநிரல் நினைவகத்திலிருந்து கொண்டு வரப்படும். புறத் திட்டநிரல் நினைவகத்திலிருந்து எடுகின்றபோது $\overline{PSEN}$ சைகை அதன்படி ஸ்டரோப் சைகையாகச் செயல்படுகிறது.

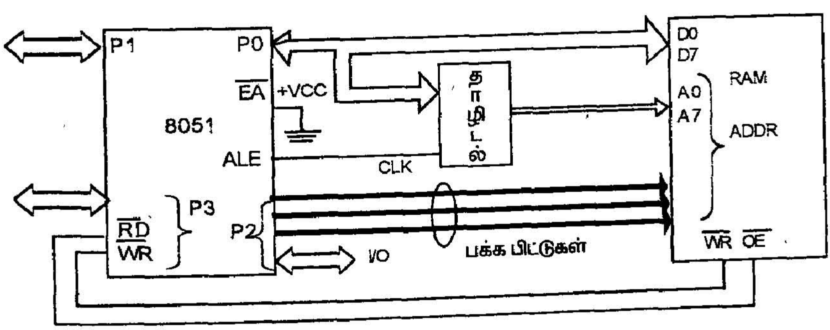
புற $\overline{RAM}$ - ஐ அடையும்போது, படி மற்றும் எழுது செயல்களைப் பொறுத்து நுண்கட்டுப்படுத்தி RD மற்றும் $\overline{WR}$ சைகைகளை உருவாக்குகிறது. "MOVX" என்கிற கட்டளையைப் பயன்படுத்திப் புற தரவு நினைவகத்தை அடையலாம். 8051-ற்கான உள்அகத் தரவு நினைவக இருப்பிடம் மூன்று கட்டமாக பிரிக்கப்பட்டுள்ளது. தாழ் 128 பைட்டுகள், உயர் 128 பைட்டுகள் மற்றும் SFR- கள் உயர் முகவரிகள் மற்றும் SFR- கள் ஒரே முகவரி இடக்கட்டத்தை உபயோகிக்கும். 8051-ன் தரவு நினைவகப் படமம் படம் 5.9இல் காட்டப்பட்டுள்ளது.



படம் 5.9. 8051 தரவு நினைவகப் படமம்

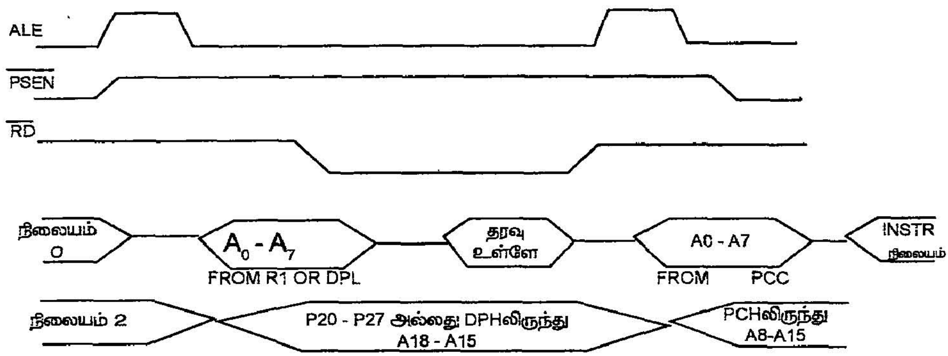
படம் 5.9இல் உயர் முகவரியை மறைமுக முகவரியால் மட்டும் அடைய முடியும் என்பதையும், SFR - கள் நேர் முகவரியால் மட்டுமே அடைய முடியும் என்பதையும் காட்டுகிறது. அதற்கு மாறாகத் தாழ் முகவரியை நேர் அல்லது மறைமுக முகவரியால் அடைய முடியும்.

படம் 5.10இல் புறத் தரவு நினைவகத்தை இணைப்பதற்கான சுற்று காட்டப்பட்டுள்ளது. துறைமுகம் 0-வால் கொடுக்கப்பட்ட பல்வின் ஒன்றாக்கி முகவரி / தரவு ஊர்தி புறத் தாழிடல் மற்றும் ALE சைகையால் ஒன்றின் பலவாக மாற்றப்படும். துறைமுகம் 2 உயர் வரிசை முகவரி ஊர்தியைக் கொடுக்கும். 8051லிருந்து RD மற்றும் WR சைகைகள் நினைவகம், படி மற்றும் நினைவகம் எழுது என்ற செயல்களைத் தேர்வு செய்யும்.

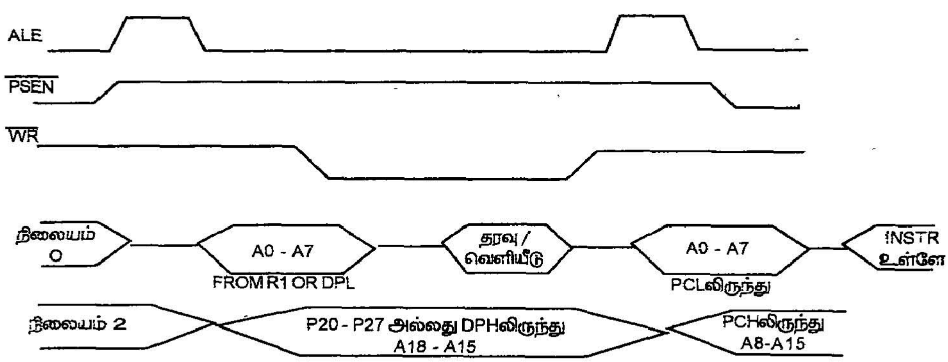


படம் 5.10 புறத்தரவு நினைவகம் அடைதல்

படம் 5.11 (அ) மற்றும் (ஆ) என்பன, புறத்தரவு நினைவகம் படி மற்றும் எழுது சுழற்சிகளுக்கான நேரமுறை அலைவடிவங்களாகும்.



படம் 5.11. (அ) புறத் தரவு நினைவகம் படி சுழற்றிக்கான நேரமுறை அலைவடிவம்



படம் 5.11 (ஆ) புறத் தரவு நினைவகம் எழுது சுழற்சிக்கான நேரமுறை அலைவடிவம்

புற நினைவகத்தின் முகவரி ஒன்று அல்லது இரண்டு பைட்டுகளைக் கொண்டிருக்கலாம். ஒரு பைட் அளவு கொண்ட முகவரிகள், ஒன்று அல்லது அதற்கு மேற்பட்ட I/O கட்டுப்பாட்டு வரிகளுடன் சேர்ந்து புற RAM-ஐ அடைய வழி செய்கிறது. உயர் முகவரி பைட் செல்கின்றபோது, இரண்டு பைட் அளவு கொண்ட முகவரிகள் பயன்படுத்தப்படும்.

புற நினைவகத்தை இரண்டு முறைகளில் அடையலாம். அவை யாவன;

i) புறத் திட்டநிரல் நினைவகத்தின் மூலம் அடையலாம்.

ii) புறத் திரவு நினைவகத்தின் மூலம் அடையலாம்.

புறத் தரவு நினைவகத்தை அடைவதற்கான கட்டளைகள் அட்டவணை 5.3இல் விளக்கப்பட்டுள்ளன.

அட்டவணை 5.1 புறத் தரவு நினைவகத்தை அடைவதற்கான கட்டளைகள் :

செயற்குறிப்பு	செயல்
MOVX A, @ Rp	Rp-இல் உள்ள புறமுகவரியை A இல் பிரதி எடு
MOVX A, DPTR	DPTR - லுள்ள புற முகவரியை A க்குப் பிரதி எடு
MOVX A, Rp, A	தரவினை A-லிருந்து Rp-லுள்ள புற முகவரிக்குப் பிரதி எடு
MOVX A, DPTR, A	தரவினை A-லிருந்து DPTR - லுள்ள புற முகவரிக்குப் பிரதி எடு

புற நினைவகத்தை அடைய முக்கியமான விபரங்களைக் கருத்தில் கொள்ள வேண்டும்.

- அனைத்துப் புறத் தரவும் புற ROM உடன் நகர்த்தலாம் அல்லது A பதிவேடு சம்மந்தப்பட்ட புற RAMஇன் மூலம் நகர்த்தலாம்.
- புற நினைவகத்தை அடையும்பொழுது, RP 256 பைட்டுகளை முகவரியிட முடியும் மற்றும் DPTR 64 கி பைட்டுகளை முகவரியிட முடியும்.
- MOVX கட்டளை, புற ROM அல்லது I/O முகவரிகள் 98ஐ அடையப் பயன்படும்.
- PCஐ பயன்படுத்தி புற ROM ஐ அடைய வேண்டுமெனில், அதனை A உடன கூட்டுவதற்கு முன்பாக, அது 1ஆல் உயர்த்தப்படும்.

5.9 உள்அக : ROM (Internal ROM)

8051இல் தரவு நினைவகமும், திட்டநிரல் குறியீடு நினைவகமும் இரு வேறுபட்ட வெளிப்படையான நினைவகங்களாக ஒழுங்காக அமைக்கப்பட்டுள்ளன. ஒவ்வொன்றும் ஒரே மாதிரியான முகவரி நெடுக்கங்களைப் பெற்றிருக்கும்.

உள்அக ROM-இல் அடங்கியுள்ள உள்அகத் திட்டநிரல் குறியீட்டின் கட்டத்தில் குறியீடு முகவரி 0000H முதல் 0FFFH வரை உபயோகத்திலிருக்கும். முகவரித் திட்டநிரல் குறியீட்டு பைட்டுகளின் முகவரிகள் 0000H முதல் FFFFH வரை, சாதாரணமாக PCஇல் பயன்படுத்தப்படும். திட்டநிரல் முகவரி வரி 0FFFHக்கு மேலாகயிருந்தால், அது உள்அக ROMஇன் கொள்ளளவிற்கு அதிகமாயிருந்தால், 8051 தன்னிச்சையாக புறத் திட்டநிரல் நினைவகத்திலிருந்து குறியீட்டு பைட்டுகளை எடுத்துக்கொள்ளும். புற நினைவக முகவரிகளிலிருந்து (0000H - FFFFH) முகவரிகள்புற அடைதல் முனை மூலம் (DIPன் மேலுள்ள EA முனை 31) தரையிட்டுக் குறியீட்டு பைட்டுகளைத் தனிப்பட்ட வகையில் எடுத்துக்கொள்ளலாம். குறியீடு எங்கிருக்கிறது என்பதைப் பற்றி PC கவலைப் படுவதில்லை. சுற்றுத் திட்டமிட்டாளர், அந்தக் குறியீடு முழுவதுமாக உள்அக ROMஇல் காணப்பட வேண்டுமா? முழுவதுமாக வெளிப்புற ROM அல்லது உள்அக மற்றும் வெளிப்புற ROM இரண்டிலும் சேர்ந்தே காணப்பட வேண்டுமா? எனத் தீர்மானிக்கலாம்.

RAM (தரவு நினைவகம்) :

32 முதல் 47 வரையுள்ள 16 பிட் அளவு கொண்ட இருப்பிடங்கள், பிட் மற்றும் பைட் ஆகிய இரண்டு முறைகளிலும் முகவரியிடக் கூடிய தன்மை கொண்ட இருப்பிடங்களாகும். இவைகள் 16X8=128 பிட் இருப்பிடங்களைக் கொண்டிருக்கும். இவைகளைத் தனித்தனியாகப் பிட் முறையில் முகவரியிடலாம். இதனை அடுத்து அமைந்துள்ள 80 இருப்பிடங்களையும், பைட் முறையில் மட்டுமே முகவரியிட முடியும். பிட் முறையில் முகவரியிட முடியாது.

நுண்கட்டுப்படுத்தியில் 21 SFRகளும், SFR பகுதியில் அமையப்பெற்றிருக்கும். இதில் 11 SFRகளை மட்டும் பிட் மற்றும் பைட் ஆகிய இரண்டு முறைகளிலும் முகவரியிடலாம். உள் அகத் தரவு நினைவகத்திலுள்ள முகவரிகள், எப்பொழுதும் ஒரு பைட் நீளம் கொண்டதாக இருக்கும்.

புற RAM நினைவகத்தைப் பயன்படுத்தும் போது $\overline{RD}$ மற்றும் $\overline{WR}$ என்கிற சைகைகளை CPU உருவாக்குகிறது. இதன்மூலம் புறத் தரவு நினைவகத்திலிருந்து படிக்கப்படுகிறது அல்லது புறத் தரவு நினைவகத்தில் எழுதப்படுகிறது. புற தரவு நினைவகத்தின் கொள்திறன், அதிகபட்சமாக 64 கி பைட் இருக்கலாம். பொதுவாக, நினைவகத்திற்குரிய முகவரி, ஒரு பைட் அல்லது இரண்டு பைட் நீளம் கொண்டதாக இருக்கும்.

மறைமுக முகவரியிடும் முறையைப் பயன்படுத்தி மட்டும்தான், புறத் தரவு நினைவகத்தை அடைய முடியும். ஒரு பைட் முகவரி நீளம் கொண்ட கட்டளைகளில், அதன் முகவரி R0 மற்றும் R1 என்கிற பதிவேடுகளின் மூலம் மறைமுகமாக குறிப்பிடப்படுகிறது.

இரண்டு பைட் முகவரி நீளம் கொண்ட கட்டளைகளில், அதன் முகவரி DPTR என்கிற பதிவேட்டின்மூலம் மறைமுகமாகக் குறிப்பிடப்படுகிறது. புறத் தரவு நினைவகப் பயன்பாட்டின்போது, சேமக்கலம் எப்பொழுதும் தரவு இருக்கின்ற இயமியாக அல்லது தரவு சென்றடைகின்ற இயமியாகச் செயல்படும். MOVX என்கிற கட்டளையின்மூலம் மட்டும்தான் புற RAMஐ அடைய முடியும்.

8051 நுண்கட்டுப்படுத்தி 256 பைட்அளவு கொண்ட உள்அகத் தரவு நினைவகத்தைக் கொண்டிருக்கும். உபயோகத்திற்கேற்ப நுண்கட்டுப்படுத்தியுடன் 64K பைட் அளவு கொண்ட புறத் தரவு நினைவகத்தை இணைத்துக்கொள்ளலாம்.

நுண்கட்டுப்படுத்தியுடன் புறத் தரவு நினைவகம் இணைக்கப்பட்ட சுற்று, படம் 5.10இல் கொடுக்கப்பட்டுள்ளது. இதில் ALE சைகை, தாழிடல் சுற்றைச் செயலியக்கம் செய்வதற்குப் பயன்படுகிறது.

புறத் தரவு நினைவகத்தைப் பயன்படுத்த MOVX வகையைச் சார்ந்த கட்டளைகள் பயன்படுத்தப்படுகின்றன. RD மற்றும் WR ஆகிய சைகைகள், புறத் தரவு நினைவகத்தின் படி மற்றும் எழுது ஸ்ட்ரோப் சைகைகளாகச் செயல்படும்.

உள்அக நினைவகம் : (Internal Memory)

செயல்பாட்டிலுள்ள கணிப்பொறியில், திட்டநிரல் குறியீட்டுப் பைட்டுகளுக்காக நினைவகம் கண்டிப்பாக இருக்கும். 8051இல் இரு உட்புற RAM மற்றும் ROM நினைவகம் உள்ளன. பொதுவாகத் திட்டநிரல் இயக்கத்தின்போது, ROM மற்றும் RAM நினைவகத்திலுள்ள மாறுகின்ற தரவினை மாற்றமுடியும் . இந்தச் செயல்களைச் செய்வதற்குக் கூடுதல் நினைவகத்தைத் தேவையான சுற்றுகளைப் பயன்படுத்தி வெளிப்புறமாக இணைக்கலாம்.

வான் நியூமான் கட்டுமானங்களுடைய (Von Neumann architectures) நுண்கட்டுப்படுத்தியில் தனி நினைவக முகவரி , திட்டநிரல் குறியீடு அல்லது தரவிற்காகப் பயன்படும். ஆனால், இரண்டுக்குமல்ல. 8051 (Hazard architecture) ஹசார்ட் கட்டுமானத்தைக் கொண்டது. இதில் ஒரே முகவரி, வெவ்வேறு நினைவகங்களில் குறியீடு மற்றும் தரவிற்காகப் பயன்படுத்தப்படும்.

உள்அக RAM (Internal RAM) :

128 பைட் உள்அக RAMஇன் விரிவான படம் 5.7-இல் காட்டப்பட்டுள்ளது. இது மூன்று வேறுபட்ட பகுதியாக ஒழுங்காக அமைக்கப்பட்டுள்ளது.

1. 00_H முதல் $1F_H$ வரை 32 பைட்டுகள் 32 வேலை செய்யும் பதிவேடுகளாக நிர்மாணிக்கப்பட்டு, நான்கு வைப்பகங்களாக ஒழுங்காக அமைக்கப்பட்டுள்ளன. ஒவ்வொன்றும் எட்டு பதிவேடுகளைக் கொண்டுள்ளன. நான்கு பதிவேடு வைபகங்கள் O முதல் 3 வரை குறிக்கப்பட்டு, ஒவ்வொன்றும் R0 முதல் R7 வரை எட்டுப் பதிவேடுகளை வைத்திருக்கும். ஒவ்வொரு பதிவேடும் பெயரால் குறிக்கப்படலாம். (அதன் வைப்பகம் தேர்வு செயய்யப்படும்போது) அல்லது அவற்றின் RAM முகவரியின் மூலம் குறிக்கப்படலாம். அதாவது வைப்பகம் 3இன் RO என்பது RO (வைப்பகம் 3 தற்சமயம் தேர்வு செய்யப்படும்போது) அல்லது முகவரி 18H (வைப்பகம் தேர்வு செய்தாலும் அல்லது செய்யவில்லையென்றாலும்), எந்த நேரத்தில் திட்டநிரல் செயல்பட்டுக் கொண்டிருந்தாலும் PSW உள்ள பிட்டுகள், RSO மற்றும் RS1 எந்த வைப்பகத்தின் பதிவேடுகள் தற்சமயம் பயன்பாட்டில் உள்ளன என்பதைக் கணக்கிடும். தேர்வு செய்யப்படாத பதிவேடு வைப்பகங்கள், பொது நோக்கு RAM ஆகப் பயன்படும். மறு அமைவின்போது, வைப்பகம் O தேர்வு செய்யப்படும்.

2. ஒரு பிட் முகவரியிடத்தக்க பரப்புடைய 16 பைட்டுகள், 20H முதல் 2FH வரையுள்ள RAM பைட் முகவரிகளை உபயோகிக்கும். இவை 128 முகவரியிடத்தக்க பிட்டுகளை உருவாக்குகின்றன. முகவரியிடக்கூடிய பிட்டை அதனுடைய பிட் முகவரி OOH முதல் 7FH வரை குறிப்பிடலாம் அல்லது 20H முதல் 2FH வரையுள்ள எந்த பைட் முகவரியிலும் 8 பிட்டுகள் உருவாக்கலாம். அதாவது உதாரணமாகப் பிட் முகவரி 4F என்பது, பிட் 7 இன் பைட் முகவரி 29H ஆகும். திட்டநிரலுக்குத் தேவையான போது, ஒரு ஈரடி நிகழ்வினை நினைவுகூரும் பொருட்டு, முகவரியிடத்தக்க பிட்டுகள் பயன்படுகின்றன.

3. ஒரு பொது நோக்குடைய RAM பிட் பரப்பிற்கு மேல் 30H முதல் 7FH வரை முகவரியிடக்கூடியது.

5.10. திட்டநிரல் எண்ணி மற்றும் தரவுக் குறிமுள் : (Program Counter and data pointer)

8051இல் இரு 16 பிட் பதிவேடுகள் உள்ளன. அவை யாவன: திட்டநிரல் எண்ணி (Program counter (PC)) மற்றும் தரவுக் குறியீடு (DPTR). ஒவ்வொன்றும் நினைவகத்தில் முகவரியின் பைட்டை வைத்திருக்கும்.

திட்டநிரல் கட்டளை பைட்டுகள் நினைவக இருப்பிடங்களிலிருந்து கொண்டு வரப்பட்டு, அவை திட்டநிரல் எண்ணியால் குறிக்கப்படும். ROM திட்டநிரல் சில்லில் முகவரி 0000_H முதல் $0FFF_H$ வரை இருக்கலாம். $0FFF_H$-க்கு அதிகப்படியான முகவரிக்காகச் சில்லிற்கு வெளியே அல்லது 0000_H முதல் $FFFF_H$ வரை அனைத்து முகவரிக்காகவும், முழுவதுமாகச் சில்லிற்கு வெளியே இருக்கலாம். ஒவ்வொரு கட்டளை பைட் எடுக்கும்போது, PC தன்னிச்சையாக அதிகரிக்கப்படும் மற்றும் சில கட்டளைகளால் மாற்றப்படலாம். உள்அக முகவரி இல்லாத ஒரே பதிவேடு, திட்டநிரல் எண்ணி ஆகும்.

DPTR பதிவேடு இரு 8 பிட் பதிவேடுகளால் ஆனது. அவையாவன DPH மற்றும் DPL உட்புற மற்றும் வெளிப்புறக் குறியீடு அடைவதற்காகவும், வெளிப்புறத் தரவை அடைவதற்காகவும் நினைவக முகவரிகளை நிரப்புவதற்கு இவைகள் பயன்படுகின்றன. DPTR திட்டநிரல் கட்டளைகளின் கட்டுப்பாட்டிலிருக்கும். அவை அதனுடைய 16 பிட் பெயரால் குறிப்பிடப்படலாம் அல்லது ஒவ்வொரு தனிப்பட்ட பைட் DPH மற்றும் DPL பெயரால் குறிப்பிடப்படலாம். DPTRக்கென்று தனிப்பட்ட உட்புற முகவரி கிடையாது. DPH மற்றும் DPL இரண்டும் ஒவ்வொரு முகவரியுடன் நியமிக்கப்படும்.

5.11. கொடிகள் மற்றும் திட்டநிரல் நிலைப்பாட்டு சொல் : Flags and Program Statue Word (PSW)

கொடிகள் என்பன 1 பிட் பதிவேடுகள். அவை சில திட்டநிரல் கட்டளைகளின் முடிவுகளைச் சேமிக்கக் கொடுக்கப்பட்டுள்ளன. மற்ற கட்டளைகள், கொடிகளின் நிபந்தனைகளைச் சோதிப்பதற்கும் நிரூபிப்பதற்கும் பயன்படக்கூடும். மேலும், கொடிகளின் நிலைகளை அடிப்படையாகக்கொண்டு முடிவெடுக்க நிர்பந்தம் செய்யும், கட்டாயப்படுத்தும். அவை எளிய முறையில் கொடிகளைக் குறிப்பிடுவதற்காகத் திட்டநிரல் நிலைப்பாட்டுச் சொல் (PSW) மற்றும் திறன் கட்டுப்பாட்டுப் பதிவேடுகள் (Power Control Register (PCON)) இவற்றினுள்ளே தொகுக்கப்பட்டிருக்கும்.

8051இல் நான்கு எண் கணக்கியல் கொடிகள் உள்ளன. கணிதச் செயல்களுக்கேற்றாற்போல் அவைகள் தன்னிச்சையாகப் பதில் கொடுக்கக்கூடியன. மூன்று பயனாளர் பொது நோக்குடைய கொடிகள். அவைகளைத் திட்டநிரலாளரின் விருப்பத்திற்கேற்றவாறு அமைவுநிலை 1 ஆகவோ அல்லது '0' எனச் சுழி ஆக்குவதோ முடியும். கணிதக் கொடிகள் சுமை (Carry (C)) துணைச் சுமை (Auxaillary Carry (AC)) பொங்கி வழி, வழிந்தோடு (overflow (OV)) ஒப்புமை (Parity (P))

நேர் பைட் முகவரி (MSB)	பிட் முகவரி						(LSB)	வன்மியம்பதிவேடு குறி
0FFH								
0F0H	F7	F6	F5	F4	F3	F2	F1	F0	B
0E0H	E7	E6	E5	E4	E3	E2	E1	E0	ACC
0D0H	D7	D6	D5	D4	D3	D2	D1	D0	PSW
0B8H	-	-	-	BC	BB	BA	B9	B8	IP
0B0H	B7	B6	B5	B4	B3	B2	B1	B0	P3
OA8H	AF	-	-	AC	AB	AA	A9	A8	IE
OAOH	A7	A6	A5	A4	A3	A2	A1	A0	P2
98H	9F	9E	9D	9C	9B	9A	99	98	SCON
90H	97	96	95	94	93	92	91	90	P 1
88H	8F	8E	8D	8C	8B	8A	89	88	TCON
80H	87	86	85	84	83	82	81	80	P0

படம் 5.12 SFR பிட் முகவரி

5.12. சிறப்புச் செயல்சார் பதிவேடுகள் : (Special Function Registers)

நுண்கட்டுப்படுத்தி, பலவகையான சிறப்புச் செயல்களைச் செய்யக்கூடிய பதிவேடுகளைக் கொண்டுள்ளது. உள்அக 128 பைட் முகவரிகள் 00H முதல் 7 FH வரை, 8051 செயல்களில் உபயோகிக்க முடியாது. அவை ஒரு தொகுப்பான சிறப்பு உட்புற பதிவேடுகளின் மூலம் செயல்படுத்தப்படும். அவை, சிறப்புச் செயல்சார் பதிவேடுகள் என்று அழைக்கப்படுகின்றன. 80 H முதல் FFH வரையுள்ள முகவரிகளைப் பயன்படுத்தி உள்அக RAM போன்று முகவரியிடக் கூடியதாகும். நுண்கட்டுப்படுத்தி 8 பிட்டுகள் உடைய 21 வகையான சிறப்புச் செயல்சார் பதிவேடுகளைக் கொண்டுள்ளன. படம் 5.12-இல் பிட் SFR முகவரி காட்டப்பட்டுள்ளது.

ACC(A), PSN, P0, P1, P2, P3, IP, IE, TCON மற்றும் SCON என்கிற 11 பதிவேடுகளும், பிட் மற்றும் பைட் முறைகளில் முகவரியிடக்கூடிய தன்மை கொண்ட சிறப்புச் செயல்சார் பதிவேடுகளாகும். மீதமுள்ள SP, DPH, DPL, TMOD, TL0, TH0, TL1, TH1, SBUF மற்றும் PCON என்கிற 10 பதிவேடுகள், பைட் முறையில் மட்டுமே முகவரியிடக் கூடிய தன்மை கொண்ட சிறப்புச் செயல்சார் பதிவேடுகளாகும். இவைகள் பிட் முறையில் முகவரியிட முடியாது.

16 பிட் DPTR பதிவேட்டில் DPH மற்றும் DPL என்பன முறையே உயர் மற்றும் தாழ் பைட் பதிவேடுகளாகும். இது புறத் தரவு நினைவகத்தினை (External Data memory) அடைய வழி செய்வதற்குப் பயன்படுகிறது. 16 பிட் T_0 நேரச்சுற்று / எண்ணி பதிவேட்டில் TH0 மற்றும் TL0 என்பன உயர் மற்றும் தாழ் பைட் பதிவேடுகளாகும். அதேபோன்று TH1 மற்றும் TL1 என்பன, 16 பிட் நேரச்சுற்று / எண்ணி பதிவேட்டின் உயர் மற்றும் தாழ் பைட் பதிவேடுகளாகும். P0, P1, P2 மற்றும் P3 என்பன, நான்கு I / O நிலையத் தாழிடல்களாகும்.

நேரச்சுற்று / எண்ணி செயல்களுக்கு TMOD மற்றும் TCON என்கிற பதிவேடுகள் பயன்படுத்தப்படுகின்றன. IE மற்றும் IP பதிவேடுகள் தடங்கல் செயல்பாடுகளுக்குப் பயன்படுத்தப்படுகின்றன. SCON, SBUF மற்றும் PCON பதிவேடுகள் தொடர் பரப்புச் செயலுக்காகவும், PCON பதிவேடு 8051இன் திறன் வகையினைத் தேர்ந் தெடுப்பதற்காகவும் பயன்படுத்தப்படுகின்றன.

SFR பகுதியிலுள்ள பயன்படுத்தப்படாத முகவரி (80H - FFH) இருப்பிடங்கள், வேறு செயலுக்காகப் பதிவு செய்யப்பட்டுள்ளது. இதனைப் பயனாளிகள் திட்டநிரலில் பயன்படுத்த முடியாது. SFR பெயர்களும் அதற்குச் சமமான உள்ளீட்டு முகவரிகளும் அட்டவணை 5.2-இல் கொடுக்கப்பட்டுள்ளது. SFR-இன் பெயர்கள் சில செயற்பாட்டுக் குறியீடுகளில் (OPCODE) அது வேலை செய்யும் பெயர்களிலேயே குறிக்கப்படும்.

அதாவது A அல்லது TH0 எனக் குறிக்கப்படும் மற்றும் அவற்றை, மற்ற செயற்பாட்டு குறியீடுகளின் முகவரியின் மூலமாகவும் குறிப்பிடலாம். அதாவது 0E0H அல்லது 8CH எனக் குறிப்பிடலாம். SFR களை அவற்றிற்குரிய பெயர்களைக் கொண்டோ அல்லது அவற்றிற்குரிய முகவரிகளைக் கொண்டோ அடைய, வழி செய்துகொள்ளலாம்.

PC, SFR-இன் ஒரு பகுதியல்ல. அது உள்ளீட்டு முகவரியினைப் பெற்றிருக்காது.. குறிப்பாகத் திட்டநிரலில் பயன்படுத்தக்கூடிய எந்த முகவரியும், கண்டிப்பாக ஒரு எண்ணைக்கொண்டு தொடங்க வேண்டும். 0 உடன் தொடங்கும் A, SFRக்கான முகவரி E0H என இப்படியாகவும் இருக்கலாம். இந்த எண்ணைப் பயன்படுத்தாமல் போனால், திட்டநிரலைத் திரட்டும்போது கூட்டாளர் தவறு (assembler error) ஏற்படும்.

அட்டவணை 5.2 SFR-இன் பெயர்கள் மற்றும் அதற்குச் சமமான உள்ளீடு RAM முகவரிகள்

பெயர்	செயல்பாடு	பைட் முகவரி
A*	சேமக்கலம்	0E0H
B*	B பதிவேடு (எண் கணிதம்)	0F0H
DPH	DPTR -இன் உயர் பைட் (புற நினைவகத்தை முகவரியிட)	83H
DPL	DPTR இன் தாழ் பைட் (புற நினைவகத்தை முகவரியிட)	82H
IE*	துடங்கல் செயலியக்கக் கட்டுப்பாடு	0A8H
IP*	துடங்கல் முன்னுரிமைக் கட்டுப்பாடு	0B8H
P0*	உள்ளீடு / வெளியீடு துறைமுகத் தாழிடல் (துறைமுகம் 0)	80H
P1*	உள்ளீடு / வெளியீடு துறைமுகத் தாழிடல் (துறைமுகம் 1)	90H
P2*	உள்ளீடு / வெளியீடு துறைமுகத் தாழிடல் (துறைமுகம் 2)	A0H
P3*	உள்ளீடு / வெளியீடு துறைமுக தாழிடல் (துறைமுகம் 3)	B0H
PCON	திறன் கட்டுப்பாடு	87H
PSW*	திட்டநிரல் நிலைப்பாட்டுச் சொல்	0D0H
SCON*	தொடர் கட்டுப்பாடு	98H

பெயர்	செயல்பாடு	பைட் முகவரி
SBUF	தொடர் தரவுத் தாங்கி	99H
SP	அடுக்குக் குறிமுள்	81H
TMOD	நேரமானி / எண்ணி வகை கட்டுப்பாடு	89H
TCON*	நேரமானி / எண்ணி கட்டுப்பாடு	88H
TL0	நேரமானி 0 தாழ் பைட்	8AH
TH0	நேரமானி 0 உயர் பைட்	8CH
TL1	நேரமானி 1 தாழ் பைட்	8BH
TH1	நேரமானி 1 உயர் பைட்	8DH

✪ குறியிட்ட SFRகள், பிட் மற்றும் பைட் ஆகிய இரு முறைகளிலும் முகவரியிடக் கூடியவை.

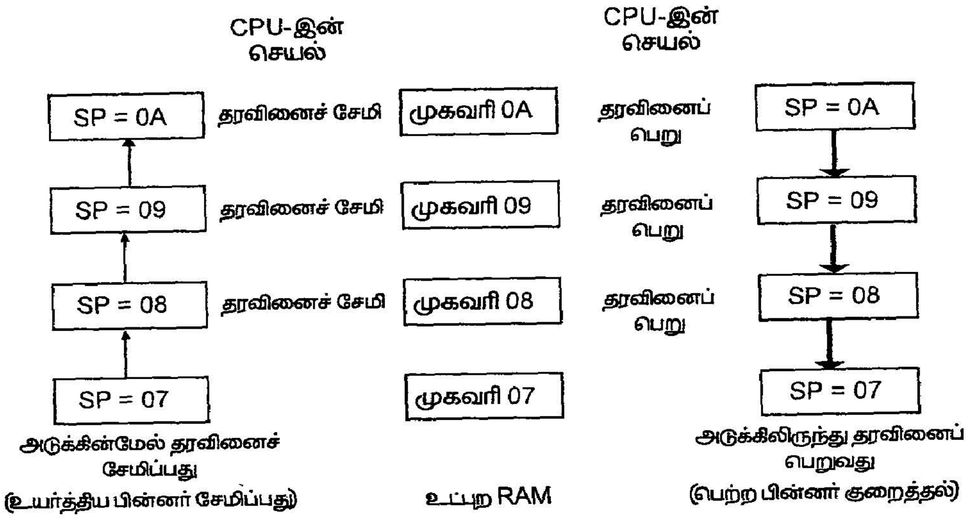
5.13 அடுக்கு மற்றும் அடுக்குக் குறிமுள் : (Stack and the Stack Pointer)

அடுக்கு என்பது, உட்புற RAMஇன் ஒரு பரப்பு. இது சில செயல்பாட்டுக் குறியீடுகளுடன் சேர்ந்து, விரைவாகத் தரவினைச் சேமிப்பதற்கும் எடுப்பதற்கும் பயன்படும். 8 பிட் அடுக்குக் குறிமுள் (SP Stack Pointer), பதிவேடு 8051இல் பயன்படுத்தப்படுகிறது. இது உட்புற RAM இன் முகவரியை வைத்திருக்கும். அதற்கு அடுக்கின் உச்சி (top of the stack) என்று பெயர். SP பதிவேட்டிலுள்ள முகவரி என்பது, உட்புற RAMஇன் இருப்பிடம். அங்கே தரவின் கடைசி பைட், அடுக்குச் செயல்பாட்டின் போது சேமிக்கப்படும்.

அடுக்கில் தரவினை வைக்கும்போது, தரவினை அடுக்கினில் சேமிப்பதற்கு முன்பாக SP உயர்வடையும். ஆகையால், அடுக்கானது சேமிக்கப்பட்ட தரவாக உயரும். அடுக்கிலிருந்து தரவினை மறுபடியும் எடுக்கும்போது, அடுக்கிலிருந்து பைட் படிக்கப்படும் மற்றும் சேமிக்கப்பட்டுள்ள தரவின் அடுத்துள்ள இடத்திற்கு SP குறைந்து அதனைக் குறிக்கும்.

அடுக்கின் செயல்பாடு மற்றும் SP படம் 5.13இல் காட்டப்பட்டுள்ளது. 8051இன் மறுஅமைவின்போது, SP 07Hஇல் அமையும். அதனை உட்புற RAM முகவரியின் எந்த இடத்தில் வேண்டுமானாலும், திட்டமிட்டாளரால் தரவு நகர்க் கட்டளையைப் பயன்படுத்தி மாற்ற முடியும்.

உட்புற RAMஇன் அளவைவிட அடுக்கின் உயரம் வரம்புக்குட்பட்டது. பதிவேடு வைப்பகம், பிட் முகவரியிடக்கூடிய RAM மற்றும் கீறலட்டை RAM பரப்புகளின் மீது அரிய தரவினை எழுதுவதற்கான திறனை அடுக்கப் பெற்றுள்ளது. முன்னதாக வரையறுக்கப்பட்ட அடுக்கு எல்லைகளுக்கு அப்பால் விருத்தி செய்யாமல் நிர்மாணிப்பது, திட்டநிரலாளரின் பொறுப்பாகும்.

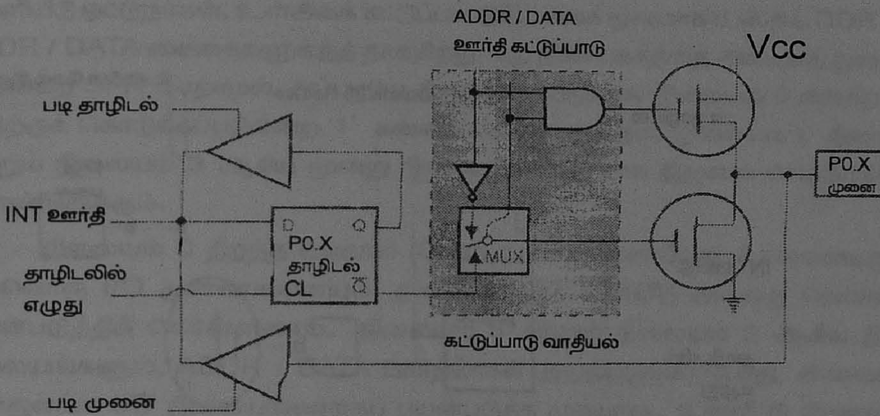


படம் 5.13 அடுக்கின் செயல்பாடு

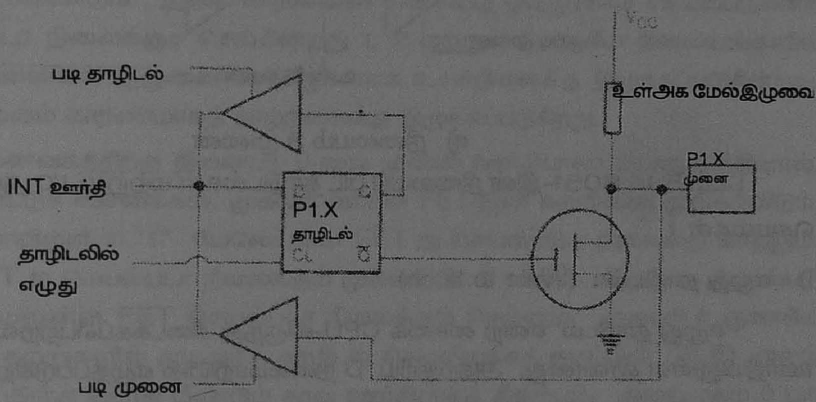
சாதாரணமாக அடுக்கு ஆனது உள்அக RAM-இன் உயர்வான இடத்தில், பதிவேடு, பிட் மற்றும் கீறலட்டை உள்அக RAM பரப்புகளின் முரண்பாடுகளைத் தவிர்ப்பதற்காக, SP பதிவேட்டில் வைக்கப்பட்டுள்ள தேர்ந்தெடுக்கப்பட்ட தகுந்த எண்ணால் வைக்கப்பட்டிருக்கும்.

5.14 உள்ளீடு / வெளியீடு நிலையங்கள் : (I / O Ports)

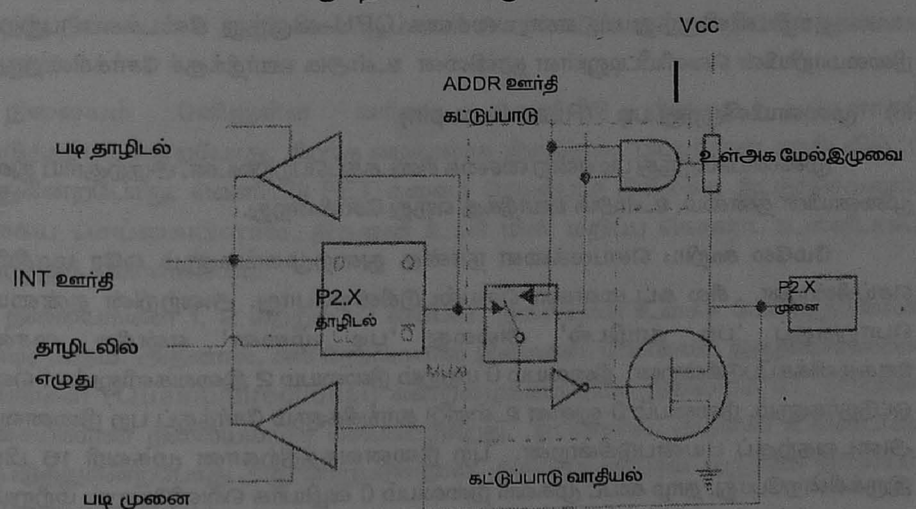
நுண்கட்டுப்படுத்தி 8051 நான்கு உள்ளீடு / வெளியீடு நிலையங்களை, இடைப்பரிமாறலுக்காகக் கொண்டிருக்கும் ஒவ்வொரு நிலையமும், எட்டு வரிகளைக் கொண்டிருக்கும். நான்கு நிலையங்களும், இருதிசையில் (bidirectional) செயல்படுகின்ற தன்மை கொண்டவை. ஒவ்வொரு நிலையமும், ஒரு தாழிடல் (P0 முதல் P3 வரையுள்ள சிறப்புச் செயல்சார் பதிவேடுகள்), ஒரு வெளியீடு ஓட்டுநர் (Output driver) மற்றும் ஒரு உள்ளீடு தாங்கி (Input buffer) ஆகியவற்றைக் கொண்டிருக்கும். நான்கு நிலையங்களின் அமைப்பும் படம் 5.14இல் காட்டப்பட்டுள்ளது.



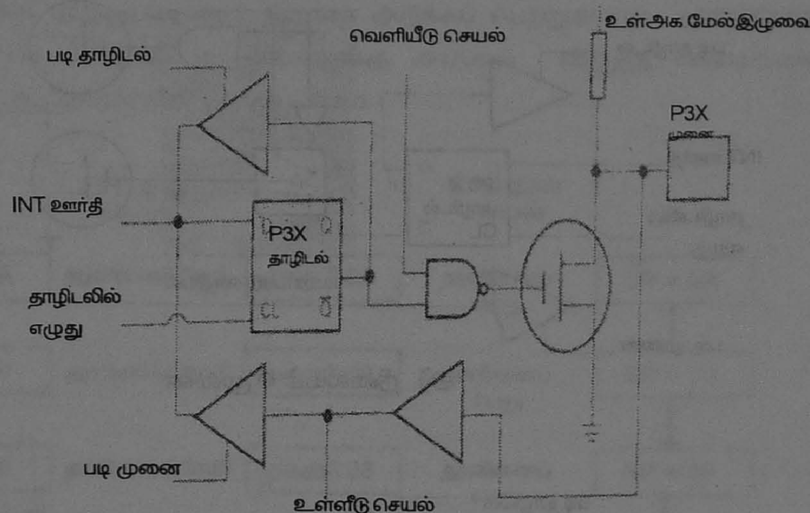
அ) நிலையம் 0 முனை



ஆ) நிலையம் 1 முனை



இ) நிலையம் 2 முனை



ஈ) நிலையம் 3 முனை

படம் 5.14 8051–இன் நிலையப் பிட் தாழிடல்கள் மற்றும் I/O தாங்கிகள்

செயல்கள் :

i) **எழுது தாழிடல் :** (Write to latch)

'எழுது தாழிடல்' என்ற சைகை CPU-லிருந்து கிடைக்கப்பெற்றவுடன், உள் அக ஊர்தியிலுள்ள தரவானது, அதற்குரிய D நிலைமாறியில் எழுதப்படுகிறது.

ii) **தாழிடலிலிருந்து படி :** (Read from latch)

தாழிடலிலிருந்து படி என்ற சைகை CPU-லிருந்து கிடைக்கப்பெற்றவுடன், நிலைமாறியின் வெளியீட்டிலுள்ள தரவினை உள்அக ஊர்திக்குச் சேர்க்கின்றது.

iii) **முனையிலிருந்து படி :** (Read from pin)

முனையிலிருந்து படி என்ற சைகை கிடைக்கப்பெற்றவுடன், அதற்குரிய துறைமுக முனையின் தன்மை, உள்அக ஊர்திக்கு வந்து சேர்கின்றது.

மேலே கூறிய செயல்களை நான்கு துறைமுகங்களும் ஒரே மாதிரியாகச் செய்கின்றன. சில கட்டளைகள் செயல்படுகின்றபோது, அவற்றின் தன்மையைப் பொறுத்துப் **'படி தாழிடல்'** அல்லது **'படி முனை'** என்கிற சைகைகள் ஊக்குவிக்கப்படுகின்றன. நிலையம் 0 மற்றும் நிலையம் 2 இவைகளிலுள்ள வெளியீடு ஓட்டுநர்களும், நிலையம் 0-லுள்ள உள்ளீடு தாங்கிகளும் சேர்த்துப் புற நினைவகத்தை அடைவதற்குப் பயன்படுகின்றன. புற நினைவகத்திற்கான முகவரி 16 பிட் ஆக இருக்கின்றபோது, தாழ் பைட் முகவரி நிலையம் 0 வழியாக வெளியேறும் மற்றும் உயர் பைட் முகவரி நிலையம் 2 வழியாக வெளியேறும்.

புற நினைவகத்தை அடைய நிலையம் 0 மற்றும் நிலையம் 2 களிலுள்ள வெளியீடு ஓட்டுநர்கள், உள்அகக் கட்டுப்பாட்டுச் சைகை மூலமாக உள்அக DOR மற்றும் ADDR / DATA சைகைகளுக்குத் தாவுகிறது. புற நினைவகத்தை அடையத் துறைமுகம் 2 என்கிற SFR மாறாமல் அப்படியே இருக்கும். ஆனால் நிலையம் 0 என்கிற SFR, அதற்குக் கொடுக்கப்படுகின்ற '1' களைக் கொண்டிருக்கும். நிலையம் 1, நிலையம் 2 மற்றும் நிலையம் 3 ஆகிய மூன்று நிலையங்களும், அக இழுவை வெளியீடுகளைக் கொண்டிருக்கும்.

நிலையம் 0 திறந்த வடிகால் (Open drain) வெளியீட்டைக் கொண்டிருக்கும். ஒவ்வொரு I/O முனைகளையும், தனித்தனியாக உள்ளீடு அல்லது வெளியீடாகப் பயன்படுத்திக் கொள்ளலாம். நிலையம் 0 மற்றும் நிலையம் 2 ஆகிய இரண்டு நிலையங்களும், ADDR / DATA ஊர்தியாகச் செயல்படுகின்றபோது, அவைகளைப் பொதுவான I/O நிலையங்களாகப் பயன்படுத்த முடியாது. உள்ளீட்டு நிலையமாகப் பயன்படுத்தப்படுகின்றபோது நிலையம் முனைக்குரிய தாழிடல் கண்டிப்பாக '1'ஐ கொண்டிருக்க வேண்டும். இதன் காரணமாக வெளியீடு ஓட்டுநராகச் செயல்படுகின்ற FET செயல்படா நிலைக்குச் செல்கின்றது 1, 2 மற்றும் 3 ஆகிய நிலையங்களில் அதற்குரிய முனைகள் இழுவை மேல் மூலமாக உயர்நிலைக்கு இழுக்கப்படுகின்றது. ஆனால் புற மூலம் காரணமாகத் தாழ்நிலைக்கு இழுக்கப்படுகிறது.

புற நினைவகத்திற்கு நிலையம் 0-வை முகவரி ஊர்தியாகப் பயன்படுத்தினால், அகக் கட்டுப்பாட்டுச் சைகைகள், முகவரி வரிகள் FET-இன் கதவினை முடுக்கிவிடும். முகவரி முனையிலுள்ள '1' மேலேயுள்ள FET ஐ செயல்படும் நிலைக்கு மாற்றும். கீழேயுள்ள FET ஐ செயல்படா நிலைக்கும் முகவரி பிட் 0 எனில், கீழுள்ள செயல்படு நிலைக்கும், மேலுள்ள FET செயல்படா நிலைக்கும் செல்லும். முகவரி உருவாகிய பின்னர், புறச் சுற்றுகளில் முகவரித் தாழிடல் செயலியக்கத் துடிப்பின் (ALE) மூலம் தாழிடப்படும். அந்த ஊர்தி மீண்டும் தரவு ஊர்தியாகத் திரும்பும். துறைமுகம் 0 புற நினைவகத்திலிருந்து தரவினைப் படிக்கும். அதனால், வாதியியல் 1 தன்னிச்சையாக அனைத்து நிலையம் 0 தாழிடல்களிலும் உள்அகக் கட்டுப்பாட்டு வாதியியல் மூலமாக எழுதப்படும்.

நிலையம் 0விலுள்ள வரிகள், வெளியீடு நிலைய வரிகளாகப் பயன்படுத்தப்படுகின்றபோது, திறந்த வடிகாலாக இருக்கும். பிட்டிற்கான தாழிடலில் '1' ஐ எழுதுகின்றபோது, வெளியீடு FET களைச் செயல்படா நிலைக்கு இட்டுச்செல்லும். இத்தகைய செய்கையினால், இதனை உயர் மின் மறிப்பு கொண்ட உள்ளீடாகப் பயன்படுத்திக் கொள்ளலாம்.

நிலையங்கள் 1, 2 மற்றும் 3 என்பன, நிலையான உள்சுக மேலிழுத்தலைக் கொண்டிருக்கும். அதனால், சில நேரங்களில் இவைகள், போன்மை இருதிசையிலான நிலையங்கள் (Quasibidirectional) என்றழைக்கப்படும். நிலையம் 01, சரியான இருதிசையிலான நிலையமாகச் செயல்படுகிறது. ஏனெனில், அதனை உள்ளீடாகச் செயல்படுத்துகின்றபோது அது மாறுபடுவதால், அதற்குச் சரியான இருதிசை நிலையம் என்று சொல்லப்படுகிறது.

நிலையம் 3க்கான முனைகள், பல செயற்பாடுகள் கொண்ட நிலையங்கள் என அழைக்கப்படுகின்றன. ஏனென்றால், இவைகள் நிலைய முனைகளாக மட்டும் செயல்படாமல், வேறு செயல்களுக்காகவும் பயன்படுத்தப்படுகின்றன.

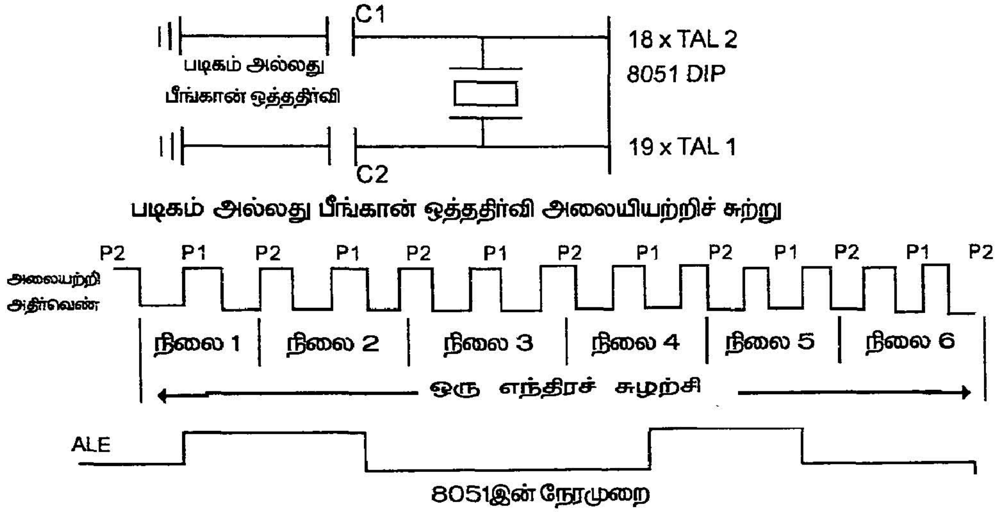
நான்கு நிலையங்களின் செயல்கள், அட்டவணை 5.3இல் சுருக்கமாகக் கூறப்பட்டுள்ளன.

அட்டவணை 5.3 நிலையச் செயல்கள்

நிலையம்	செயல்கள்
நிலையம் 0	உள்ளீடு / வெளியீடு நிலையமாகப் பயன்படுகின்றன. இரு திசை தாழ் - வரிசை முகவரி மற்றும் புற நினைவகத்திற்காகத் தரவு ஊர்தியாகப் பயன்படுகிறது.
நிலையம் 1	உள்ளீடு / வெளியீடு நிலையமாகப் பயன்படுகின்றன.
நிலையம் 2	உள்ளீடு / வெளியீடு நிலையமாகப் பயன்படுகிறது. புற நினைவகத்திற்காக உயர் வரிசை முகவரி ஊர்தியாகப் பயன்படுகிறது.
நிலையம் 3	உள்ளீடு / வெளியீடு நிலையமாகப் பயன்படும். முனை — மாற்றுப் பயன்பாடு P3.0 - RXD — தொடர் தரவு உள்ளீடு P3.1 - TXD — தொடர் தரவு வெளியீடு P3.2 - $\overline{INT0}$ — புறத் தடங்கல் 0 P3.3 - $\overline{INT1}$ — புறத் தடங்கல் 1 P3.4 - T0 — புற நேரமானி 0 உள்ளீடு P3.5 - T1 — புற நேரமானி 1 உள்ளீடு P3.6 - $\overline{WR}$ — புற நினைவக எழுது சைகை P3.7 - $\overline{RD}$ — புற நினைவகப் படி சைகை

5.15 8051-இன் அலையியற்றி மற்றும் கடிகாரம் :

8051இன் இதயமாக உள்ள இந்தச் சுற்று, கடிகாரத் துடிப்புகளை உண்டாக்கி, அனைத்து உள்அகச் செயற்பாடுகளையும் ஒத்தியங்க வைக்கின்றது. XTAL 1 மற்றும் XTAL 2 முனைகளில், ஒரு ஒத்திசைவு பின்னகத்தை இணைத்து, ஒரு அலையியற்றியைக் குவார்ட்ஸ் படிகம் மற்றும் மின்தேக்கிகளைக் கொண்டு உருவாக்கலாம். அலையியற்றியின் சுற்று மற்றும் நேரமுறையைப் படம் 5.15இல் காணலாம்.



படம் 5.15. அலையியற்றிச் சுற்று மற்றும் நேரமுறை

படிகத்தின் அதிர்வெண், நுண்கட்டுப்படுத்தியின் உள்அகக் கடிகார அதிர்வெண்ணாகும். நுண்கட்டுப்படுத்தியின் அதிர்வெண் 1 மெகாஹர்ட்ஸ் முதல் 16 மெகா ஹர்ட்ஸ் வரையிலும் இருக்கும்.

படிகம், மின்தேக்கிகள் மற்றும் தலைகீழாக்கியைக் கொண்டு அலையியற்றியை உருவாக்கிப் படிக அதிர்வெண்ணில் ஒரு துடிப்புத் தொடரை உருவாக்கலாம். நுண்கட்டுப்படுத்தியின் உள்ளே மிகச்சிறிய நேர இடைவெளியினை நிர்மானிக்கும் கடிகார அதிர்வெண் துடிப்பு (Pulse), P நேரம் எனப்படும். எளிய கட்டளை அல்லது மிகச்சிறிய இடைவெளியில் நிறைவேற்றுவது, எந்திரச் சுழற்சி ஆகும்.

செயற்குறியீடு பைட்டைக் கொணர்தல், செயற்குறியீடு (Opcode), குறியீடு பெயர்த்தல், செயற்குறியீட்டைச் செய்து முடித்தல் அல்லது ஒரு தரவு பைட்டை எழுதுதல் போன்ற நுண்கட்டுப்படுத்தியின் தனித்தனிச் செயல்பாடுகளுக்கான அடிப்படை நேர இடைவெளி நிலை என்பதாகும். ஒவ்வொரு நிலைக்கும் இரு அலையியற்றித் துடிப்புகள் வரையறுக்கப்படும்.

திட்டநிரல் கட்டளைகள் ஒன்று, இரண்டு அல்லது நான்கு எந்திரச் சுழற்சிகள் கட்டளையின் வகைப்பாட்டிற்கு ஏற்றாற்போல் நிறைவேற்றுவதற்குத் தேவைப்படும். கட்டளைகள் எடுக்கப்பட்டு, நுண்கட்டுப்படுத்தியினால் தன்னிச்சையாக நிறைவேற்றப்படும். நுண்கட்டுப்படுத்தி, முதலாவது மறு அமைவின்போது, ROM நினைவக முகவரி 0000 தொடக்கக் கட்டளையாகும்.

நேரத்தைக் கணக்கிடுவதற்கு, ஏதாவது ஒரு குறிப்பிட்ட கட்டளையை நிறைவேற்றுவதாக எடுத்துக் கொள்ளலாம். ஒரு கட்டளையை நிறைவேற்றுவதற்கான நேரம்,

$$T_{inst} = \frac{C \times 12d}{\text{படிக அதிர்வெண்}}$$

நேரச்சுற்று / எண்ணிகள் : (Timer / Counters)

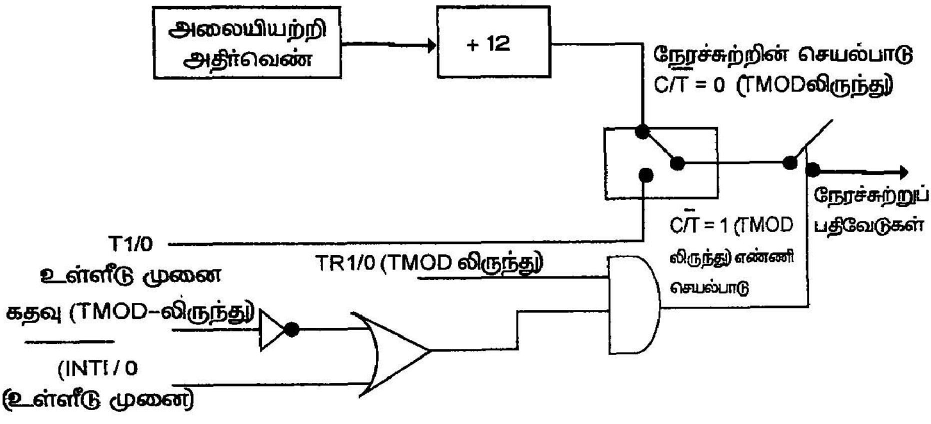
8051 இரண்டு 16 பிட் நேரச்சுற்று / எண்ணி பதிவேடுகளைக் கொண்டிருக்கும். அவையாவன, நேரச்சுற்று 0 (T0) மற்றும் நேரச்சுற்றுப் 1(T1) பதிவேடுகள் இரண்டும் நேரச்சுற்றாகவோ அல்லது எண்ணிகளாகவோ இயக்குவதற்கேற்றார்போல் அமைக்கப்பட்டிருக்கும்.

'நேரச்சுற்று' வகையில், ஒவ்வொரு எந்திரச் சுழற்சியின்போதும், நேரச்சுற்றுப் பதிவேட்டிலுள்ள மதிப்பு ஒன்று அதிகரிக்கும். அதாவது, நேரச்சுற்றாகச் செயல்படுகின்றபோது, அது எந்திரச் சுழற்சிகளை ஒவ்வொன்றாக எண்ணுகிறது. ஒரு எந்திரச் சுழற்சி 12 அலையியற்றி நேரங்களைக் (fosc) கொண்டிருக்கும். கணக்கீடு விகிதம் என்பது, அலையியற்றி அதிர்வெண்ணின் 1/12 ஆகும். உதாரணமாகப் படிக அதிர்வெண் 12 MHZ எனில், நேரச்சுற்றின் கடிகார அதிர்வெண் 1 MHZ ஆகும்.

எண்ணி வகையில், அதற்குரிய புற உள்ளீட்டு முனை T0 அல்லது T1இல் 1 லிருந்து 0 விற்கு மாறும்போது, பதிவேடு ஒன்று அதிகரிக்கும். இந்த வகையில் புற உள்ளீடு ஒவ்வொரு எந்திரச் சுழற்சியின் கட்டம் 2ன் நிலை 5-ன் (S5P2) போதும் தேர்வுச் செய்யப்படுகிறது. தேர்வுச் செய்யப்பட்டவைகள், ஒரு எந்திரச் சுழற்சியில் உயர்வாகவும் அடுத்த சுழற்சியின்போது தாழ்வாகவும் இருக்கும். அப்போது அதற்குரிய பதிவேட்டின் மதிப்பானது ஒன்று அதிகரிக்கும். இவ்வாறு அதிகரிக்கப்பட்ட புதிய கணக்கீடு, அடுத்து வருகின்ற எந்திரச் சுழற்சியின் S3P1 என்கிற நிலையின்போது பதிவேட்டில் கிடைக்கும். அதாவது, ஒரு 1லிருந்து 0 என்ற மாற்றத்தை நுண்கட்டுப்படுத்தி கண்டுபிடிப்பதற்குக் குறைந்தது இரண்டு எந்திரச் சுழற்சிகள் தேவைப்படும். அதிகபட்சக் கணக்கீடு விகிதம் ஆனது அலையியற்றியின் அதிர்வெண் களின் 1/24 மடங்காக இருக்கும். புற உள்ளீடு சைகைகளின் கடமைச் சுழற்றியில் எந்தக் கட்டுப்பாடும் கிடையாது. ஆனால், கொடுக்கப்பட்ட மட்டும் மாறுவதற்கு முன்பாக ஒரு தடவையேனும் சரிபார்க்கப்பட்டது என நிச்சயப்படுத்த வேண்டும். அது ஒரு முழுமையான எந்திரச் சுழற்றி வரையிலும் அப்படியே வைத்திருக்க வேண்டும்.

நேரச்சுற்று / எண்ணி கட்டுப்பாட்டு வாதியல் :

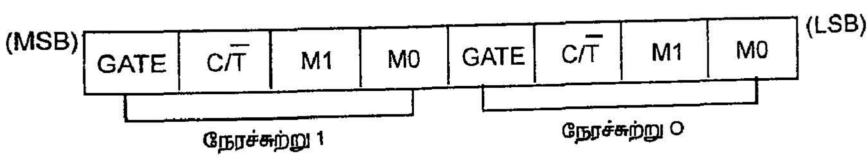
8051இன் நேரச்சுற்றுகள் மட்டும் எண்ணிகள் TCON லுள்ள TR1/0 பிட்டுகள் TMODலுள்ள கதவு பிட்டுகள் மற்றும் 8051இன் $\overline{INT1/0}$ உள்ளீடு முனைகள் ஆகியவற்றால் கட்டுப்படுத்தப்பட்டன. TMOD பதிவேட்டிலுள்ள C/$\overline{T}$ பிட் செயல்பாட்டைத் தீர்மானிக்கும். நேரச்சுற்றுக்கான தேர்வு பிட் 0 மற்றும் எண்ணி செயல்பாட்டிற்கான தேர்வு பிட் 1. இது படம் 5.16-இல் விளக்கப்பட்டுள்ளது.



படம் 5.16. நேரச்சுற்று / எண்ணி கட்டுப்பாடு வாதியல்

<u>நேரச்சுற்று 0 மற்றும் நேரச்சுற்று 1</u> :

TMOD, இந்த நேரச்சுற்றுகளில் TMOD, சிறப்புச் செயல்சார் பதிவேட்டிலுள்ள கட்டுப்பாட்டு முனைகள் எண்ணி அல்லது நேரச்சுற்று வகையைத் தேர்வு செய்கிறது. இந்த இரண்டு நேரச்சுற்று / எண்ணிகள் நான்கு செயல்பாட்டு வகைகளில் செயல்படும் தன்மை கொண்டன. எந்த வகையில் செயலாற்ற வேண்டும் என்பதை TMOD பதிவேட்டிலுள்ள M1 மற்றும் M0 பிட்டுகளைக் கொண்டு தேர்வு செய்து கொள்ளலாம்.



நேரச்சுற்று / எண்ணிகள் இரண்டிற்கும் வகைகள் 0, 1 மற்றும் 2 ஒரே மாதிரியாய் இருக்கும்.

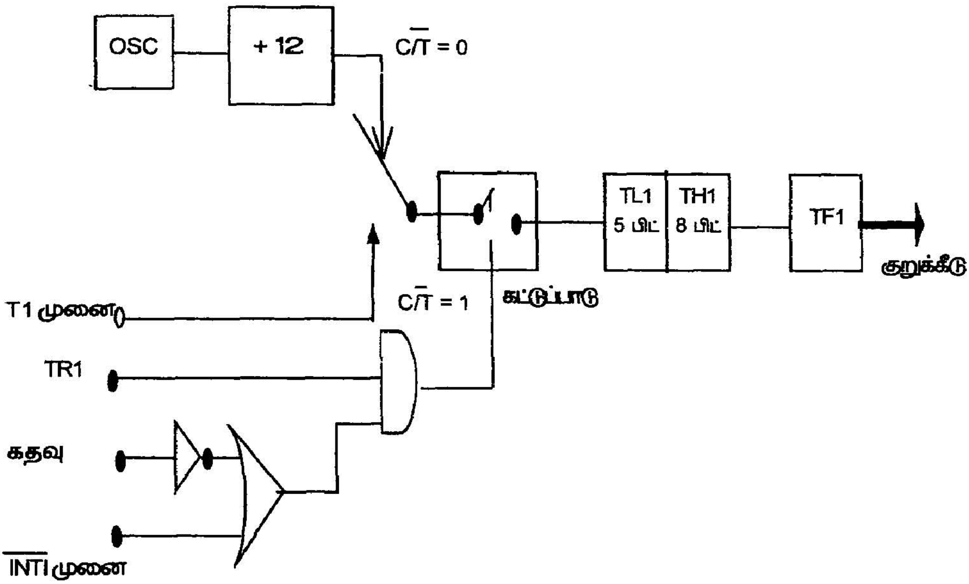
நான்கு செயல்பாட்டின் வகைகள், கீழே கொடுக்கப்பட்டுள்ளன.

M1	M0	செயலாற்றுகின்ற விதம்
0	0	8 – பிட் கொண்ட நேரச்சுற்று / எண்ணி
0	1	16 பிட் கொண்ட நேரச்சுற்று / எண்ணி
1	0	தானாகவே மறுசுமை ஆகின்ற தன்மை பெற்ற 8 பிட் கொண்ட நேரச்சுற்று / எண்ணி
1	1	(நேரச்சுற்று 0) TL0 8 பிட் கொண்ட நேரச்சுற்று / எண்ணியாகச் செயல்படும். இது நேரச்சுற்று 0-க்குரிய கட்டுப்பாட்டுப் பிட்டுகளால் கட்டுப்பாடு செய்யப்படுகிறது. TH0ம் 8 பிட் கொண்ட ஒரு நேரச்சுற்றாகச் செயல்படும். இது நேரச்சுற்று 1க்குரிய கட்டுப்பாட்டுப் பிட்டால் கட்டுப்பாடு செய்யப்படுகிறது.
1	1	(நேரச்சுற்று 1) நேரச்சுற்று / எண்ணி 1 தடுக்கப்படுகிறது.

வகை 0 : (Mode 0) (13 பிட் நேரச்சுற்று / எண்ணி)

வகை 0 யிலுள்ள இரண்டு நேரச்சுற்றுகளும் ஒரு 8 பிட் எண்ணியாகும். இந்த 13பிட் நேரச்சுற்றிற்கு MCS - 48 ஏற்றதாகும். முறை 0 இல் நேரச்சுற்றுப் பதிவேடு 13 பிட் கொண்ட பதிவேடாகச் செயல்படும். நேரச்சுற்று / எண்ணி 1–இன் முறை 0–இல் செயற்பாட்டிற்கான படம் 5.17–இல் காட்டப்பட்டுள்ளது. இந்த முறையில், நேரச்சுற்றுப் பதிவேடு, ஒரு 13பிட் பதிவேடாகும். இந்த நேரச்சுற்று செயலாற்றி முடிவடையும் போது, அனைத்தும் 1 ஆக மாறி, பின்பு அனைத்தும் 0 என மாறுகின்றபோது, TCON பதிவேட்டிலுள்ள நேரச்சுற்றுக் குறுக்கீட்டுக் கொடி TF1 அமைவு நிலையை அடையும். TMOD பதிவேட்டிலுள்ள C/T பிட்டைப் பயன்படுத்திப் பதிவேட்டை நேரச்சுற்று அல்லது எண்ணியாகச் செயல்படுத்திக் கொள்ளலாம்.

TMOD பதிவேட்டிலுள்ள C/T பிட்டைப் பயன்படுத்திப் பதிவேடு நேரச்சுற்று அல்லது எண்ணியாகச் செயல்படுத்திக் கொள்ளலாம். நேரச்சுற்று 1 எண்ணியாகச் செயல்பட வேண்டுமென்றால், TR1=1 எனவும், மற்றும் கதவு = 0 அல்லது $\overline{INT1}$ = 1 எனவும் இருக்கவேண்டும். (கதவு = 1 என அமைக்கும்போது, நேரச்சுற்றுப் புற உள்ளீடு INT1 ஆல் கட்டுப்படுத்தப்படும். இந்தச் செயல் துடிப்பின் அகலத்தைக் கணக்கிடுவதற்குப் பயன்படுகிறது.



படம் 5.17 நேரச்சுற்று / எண்ணி 1 முறை 0 : 13 பிட் எண்ணி

TCON சிறப்புச் செயல்சார் பதிவேட்டில், TRI கட்டுப்பாடு முனை ஆகவும் TMOD-இல் கதவாகவும் இருக்கும்.

TCON பதிவேடு :

TCON என்பது, 8 பிட்டுகளைக் கொண்ட ஒரு SFR பதிவேடாகும். இதனைப் பிட் மட்டும் பைட் ஆகிய இரண்டு முறைகளிலும் முகவரியிடலாம். இந்தப் பதிவேடு

நேரச்சுற்று பொங்கி வழி கொடிகள், நேரச்சுற்றுகளுக்கான ஓட்டம் கட்டுப்பாடு பிட்டுகள் (Run outrol unit), புறக் குறுக்கீடுகளுக்கான வகை கட்டுப்பாடு பிட்டுகள் மற்றும் புறக் குறுக்கீடுகளுக்கான கொடி பிட்டுகள் ஆகியவற்றினைக் கொண்டிருக்கும். TCON பதிவேட்டின் அமைப்பானது, கீழே கொடுக்கப்பட்டுள்ளது.

(MSB) D7	D6	D5	D4	D3	D2	D1	D0 (LSB)
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

குறி	இடம்	பெயர் மற்றும் செயல்கள்
TF1	TCON.7	நேரச்சுற்று 1 பொங்கி வழி கொடி. நேரச்சுற்று / எண்ணி பொங்கி வழிகின்றபோது, இது வன்மியமுறையில் அமைவு நிலைக்குச் செல்கிறது. குறுக்கீடு செயல்படுத்தப்பட்டவுடன் தானாகவே காலியாகிவிடும்.
TR1	TCON.6	நேரச்சுற்று 1 ஓட்டக் கட்டுப்பாடு முனை வன்மியத்தால் அமைவு / (மறுஅமைவு) காலி ஆகின்ற தன்மை கொண்டது. இதன்மூலம் நேரச்சுற்று / எண்ணியைச் செயற்பாடு / செயற்படா நிலைக்கு மாற்ற முடியும்.
TF0	TCON.5	நேரச்சுற்று 0 பொங்கி வழி கொடி. நேரச்சுற்று / எண்ணி பொங்கி வழிகின்றபோது, அது மென்மிய அமைவு நிலைக்குச் செல்லும். குறுக்கீடு செயல்படுத்தப்பட்டவுடன் தானாகவே காலியாகி விடும்.
TR0	TCON.4	நேரச்சுற்று 0 ஓடு கட்டுப்பாடு முனை, மென்மியத்தால் அமைவு / காலி ஆகின்ற தன்மை கொண்டது. இதன்மூலம் நேரச்சுற்று / எண்ணியைச் செயற்படு / செயற்படா நிலைக்கு மாற்ற முடியும்.
IE1	TCON.3	குறுக்கீடு 1 விளிம்புக் கொடி, புறக் குறுக்கீட்டில் விளிம்புச் சைகை கண்டுப்பிடிக்கப்படும்போது, வன்மியத்தால் அமைவு நிலையை அடையும்.
IT1	TCON.2	குறுக்கீடு 1 வகைக் கட்டுப்பாடு பிட், மென்மியம் முறையில் அமைவு நிலை / காலி செய்து கொள்ளலாம். இதன்மூலம் இறங்கு விளிம்பு / தாழ் மட்டும் தொடக்கி முறைகளில், புறக் குறுக்கீடு ஆனது எந்த முறையில் தொடங்கவேண்டும் என்பதைத் தீர்மானிக்கும்.
IE0	TCON.1	குறுக்கீடு 0 விளிம்புக் கொடி. புறக் குறுக்கீட்டில் விளிம்புச் சைகை கண்டுபிடிக்கப்பட்டால், மென்மியம் முறையில் அமைவு நிலையை அடையும். குறுக்கீடு செயல்படத் தொடங்கியவுடன் இது தானாகவே காலியாகிவிடும்.
IT0	TCON.0	குறுக்கீடு 0 வகைக் கட்டுப்பாடு பிட், மென்மியம் முறையில் அமைவு / காலி ஆக மாற்றலாம். இதன்மூலம் இறங்கு விளிம்பு / தாழ் மட்டத் தொடக்கி முறைகளில் புறக் குறுக்கீடானது எந்த முறையில் தொடங்க வேண்டும் என்பதைத் தீர்மானிக்கும்.

13 பிட் கொண்ட பதிவேடு, TH1லிருந்து 8 பிட்டுகளையும் மற்றும் TL 1-லிருந்து 5 பிட்டுகளையும் கொண்டுள்ளது. TL1இன் முதல் 3 பிட்டுகள் வரையறுக்கப்படாதவை மற்றும் அவற்றைக் கவனிக்க வேண்டியதில்லை. ஓடு கொடி (TR1)-ஐ அமைவு நிலையில் வைத்துப் பதிவேடுகளைக் காலி செய்ய முடியாது.

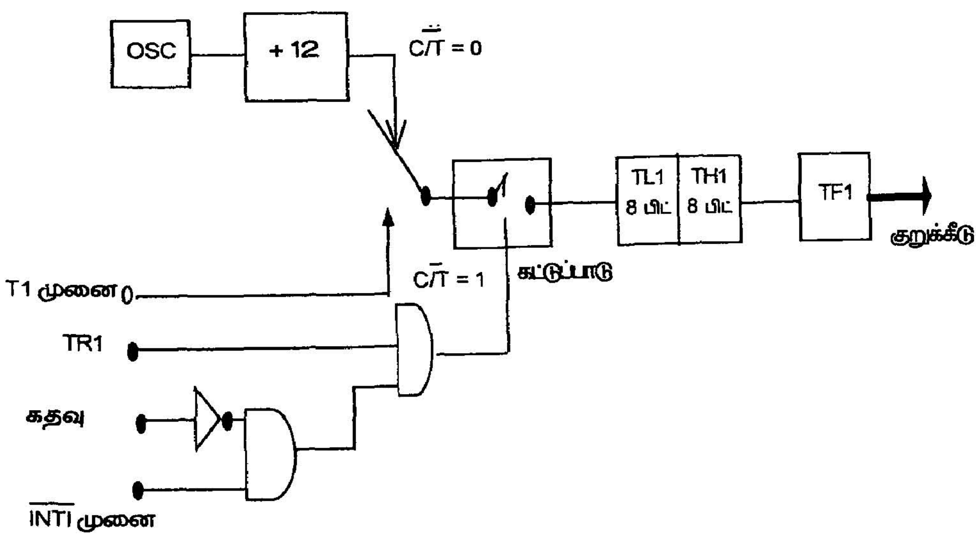
முறை 0இன் செயல்பாடு நேரச்சுற்று 0 மற்றும் 1 போன்றிருக்கும்.

TMOD பதிவேடு :

TMOD என்பது 8 பிட்டுகளைக் கொண்ட ஒரு சிறப்புச் செயல்சார் பதிவேடாகும். இதனைப் பைட் முறையில் மட்டுமே முகவரியிட முடியும். இதிலுள்ள C/T என்கிற கட்டுப்பாட்டுப் பிட்டைப் பயன்படுத்தி நேரச்சுற்றாகச் செயல்பட வேண்டுமா அல்லது எண்ணியாகச் செயல்பட வேண்டுமா என்பதைத் தேர்வு செய்துகொள்ளலாம். நான்கு முறைகளில் எந்த முறையில் செயலாற்ற வேண்டும் என்பதை, M1 மற்றும் M0 ஆகிய கட்டுப்பாடு பிட்டுகளைப் பயன்படுத்தி த்தேர்வு செய்யலாம்.

முறை 1 (16 பிட் நேரச்சுற்று / எண்ணி) :

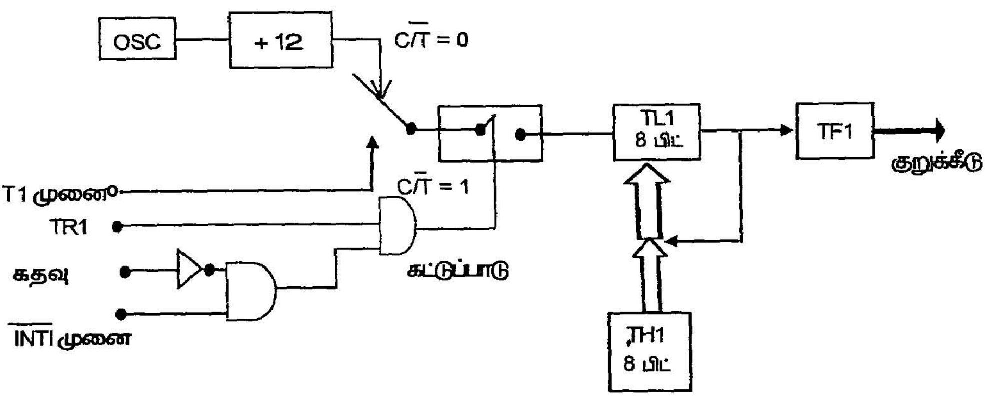
முறை 1இன் செயலானது முறை 0வைப் போன்றேயிருக்கும். ஆனால், இதில் நேரச்சுற்று 1-ற்குரிய TH1 மற்றும் TL1 ஆகிய இரண்டு 8 பிட் பதிவேடுகளும் இணைந்து, 16 பிட் பதிவேடாகச் செயல்படுகிறது. முறை 1இல் செயல்படுகின்ற நேரச்சுற்று 1க்கான செயல்விளக்கப் படமானது, படம் 5.17 இல் காட்டப்பட்டுள்ளது.



படம் 5.18 நேரச்சுற்று / எண்ணி 1 முறை 1 : 16 பிட் எண்ணி

முறை 2 (8 பிட் தன் மறு சுமை நேரச்சுற்று / எண்ணி) :

முறை 2இல் நேரச்சுற்றுப் பதிவேடானது, 8 பிட் கொண்ட பதிவேடாகச் செயல்படுகிறது. இது தானாகவே சுமையேற்றும் தன்மை கொண்டது. நேரச்சுற்று 1க்கான முறை 2ற்கான செயலின் விளக்கப்படமானது, படம் 5.19இல் காட்டப்பட்டுள்ளது. இந்த முறையில் செயல்படும். TL1 விருந்து பொங்கி வழியும்போது, TF1 மட்டும் அமைவு நிலையை அடைவதில்லை. TH1-லுள்ள மதிப்பானது TL1-இல் மறுசுமையேற்றுகிறது. மறுசுமையேற்றிய, பின்பும் TH1-இன் மதிப்பு மாறாமல் அப்படியே இருக்கும். TH1-இன் மதிப்பை மென்மியம் முறையில் மறுஅமைவு நிலைக்குக் கொண்டு வரலாம். முறை 2இன் செயல்பாடு, நேரச்சுற்று / எண்ணி 0இன் செயல்பாட்டினை ஒத்திருக்கும்.

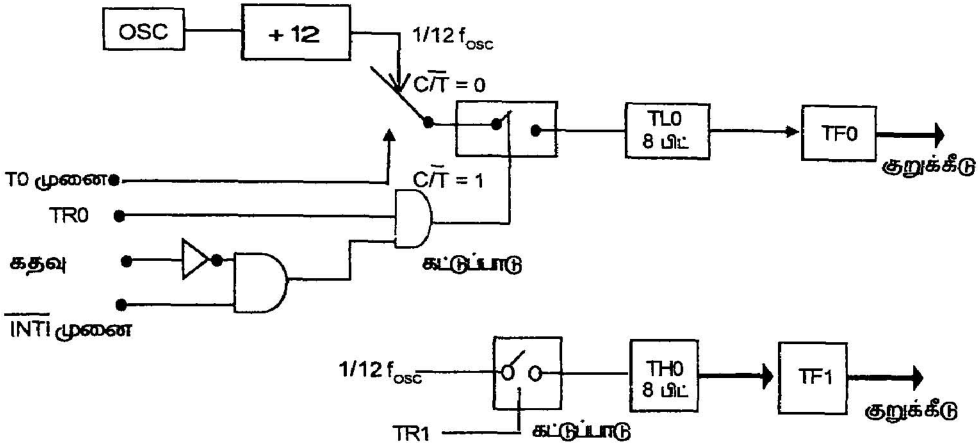


படம் 5.19. நேரச்சுற்று / எண்ணி 1 முறை 2

முறை 3 : (இரு 8 பிட் நேரச்சுற்று / எண்ணி)

முறை 3இன் நேரச்சுற்று 1 கணக்கீடை வைத்திருக்கும். TH1=0 என அமைக்கும்போது ஏற்படுகின்ற விளைவை ஒத்திருக்கும். முறை 3இன் நேரச்சுற்று 0 TL0 மற்றும் TH0 ஆகிய இரு பதிவேடுகளும் தனித்தனி எண்ணியாகச் செயல்படும். நேரச்சுற்று 0 மட்டுமே முறை 3இல் செயலாற்ற முடியும். நேரச்சுற்று 1 முறை 3இல் செயலாற்ற முடியாது. நேரச்சுற்று 0 முறை 3இல் செயலாற்றுகின்ற விளக்கப்படத்தை படம் 5.20 இல் காணலாம்.

TL0 ஆனது நேரச்சுற்று 0 இன் கட்டுப்பாடு பிட்டுகளைப் பயன்படுத்தும். நேரச்சுற்று 0க்கான கட்டுப்பாடு முனைகள் முறையே $C/\overline{T}$, GATE, $\overline{TR0}$, $\overline{INT0}$ மற்றும் TF0 அகியவை கட்டுப்பாடு சைகைகளாகச் செயல்படும். நேரச்சுற்று முறையில் TH0 பூட்டப்படும். அது நேரச்சுற்று 1லிருந்து TR1 மற்றும் TF1இன் உபயோகத்திற்கு எடுத்துக்கொள்ளும். இவ்வாறு TH0 இப்போது நேரச்சுற்று 1ஐ கட்டுப்படுத்தும். அதேபோன்று நேரச்சுற்று 1க்குரிய கட்டுப்பாடு முனைகள் $C/\overline{T}$, GATE, TR1, $\overline{INT1}$ மற்றும் TF1 ஆகியனவாகும். இவைகள் TH0 க்குரிய கட்டுப்பாடு சைகைகளாகச் செயல்படும்.



படம் 5.20. நேரச்சுற்று / எண்ணி 0 முறை 3

முறை 3இல் உபயோகத்திற்காக வேண்டிய அதிகப்படியான 8 பிட் நேரச்சுற்று அல்லது எண்ணி கொடுக்கப்பட்டிருக்கும். முறை 3இல் நேரச்சுற்று 0 உடன் இருப்பது 8051 மூன்று நேரச்சுற்று / எண்ணிகள் உள்ளதைப் போன்றதாகும். முறை 3இல் நேரச்சுற்று 0 எனில், நேரச்சுற்று 1 செயல்படு மற்றும் செயல்படா நிலைக்குத் தன்னுடைய நிலையை விட்டு மற்றும் தன்னுடைய முறைக்குத் தாவும் அல்லது தொடர் துறைமுகத்தினால் இன்னும் பயன்படுத்தக் கூடியதாக இருக்கும். நேரச்சுற்று 1க்கான குறுக்கீடு TH0இன் மூலம் கட்டுப்படுத்தப்படும். நேரச்சுற்று 0, TF1 கொடியைப் பயன்படுத்திக் கொண்டிருக்கும்போது, நேரச்சுற்று 1இன் மூலம் எவ்விதக் குறுக்கீடுகளும் ஏற்படுவதில்லை. அட்டவணை 5.4-இல் நேரச்சுற்றின் முறைகள் தொகுக்கப்பட்டுள்ளன.

அட்டவணை 4 நேரச்சுற்று முறைகளின் தொகுப்பு.

முறை	சுருக்கமான விளக்கம்
முறை 0	16 -23 பிட் நேரச்சுற்று (TL-5 பிட்டுகள் மற்றும் TH-8 பிட்டுகள்) எண்ணியின் பொங்கி வழிதல் நேரக்குறுக்கீடு கொடியினால் குறிப்பிடப்படும்.
முறை 1	16 பிட் நேரச்சுற்று (TL-8 பிட்டுகள் மற்றும் TH-8 8 பிட்டுகள்) முறை 0 வைப் போன்று மறு அமைவு பெறும் .
முறை 2	தானே இயங்குகின்ற மறுசுமை முறை, TL லிருந்து பொங்கி வழியும் 8 பிட் எண்ணி (TL-8 பிட்) TF மட்டும் அமைவு நிலைக்குக் கொண்டு செல்லாமல் TLஐயும் TH- யுங்கொண்டு மறுசுமையேற்றும்.
முறை 3	TL மற்றும் TH ஆகிய இரண்டு தனித்தனியான எண்ணிகளைக் கொண்டிருக்கும்.

5.16. குறுக்கீடு அமைப்பு : Interrrupt Structure

நுண்கட்டுப்படுத்தியின் கவனத்தை உடனடியாகப் பெறுவதற்காகக் குறுக்கீடுகள் பயன்படுத்தப்படுகின்றன. ஒரு குறுக்கீடு உருவானவுடன், நுண்கட்டுப்படுத்தி தான் செய்து கொண்டிருக்கின்ற முதன்மைத் திட்டநிரலை நிறுத்திவிட்டு, அதன் கட்டுப்பாட்டை அந்தக் குறுக்கீட்டிற்குரிய சேவை துணை நிரற்செயன்முறை இருக்கின்ற இடத்திற்குக் கொண்டு செல்கின்றது. 8051 நுண்கட்டுப்படுத்தி, ஐந்து வகையான குறுக்கீடு மூலங்களைக் கொண்டது. ஆனால் 8052 நுண்கட்டுப்படுத்தி ஆறு வகையான குறுக்கீட்டு மூலங்களைக் கொண்டதாகும். அவையாவன,

(1) புறக் குறுக்கீடு 0 (IE0)

(2) நேரச்சுற்றுக் குறுக்கீடு 0 (TF0)

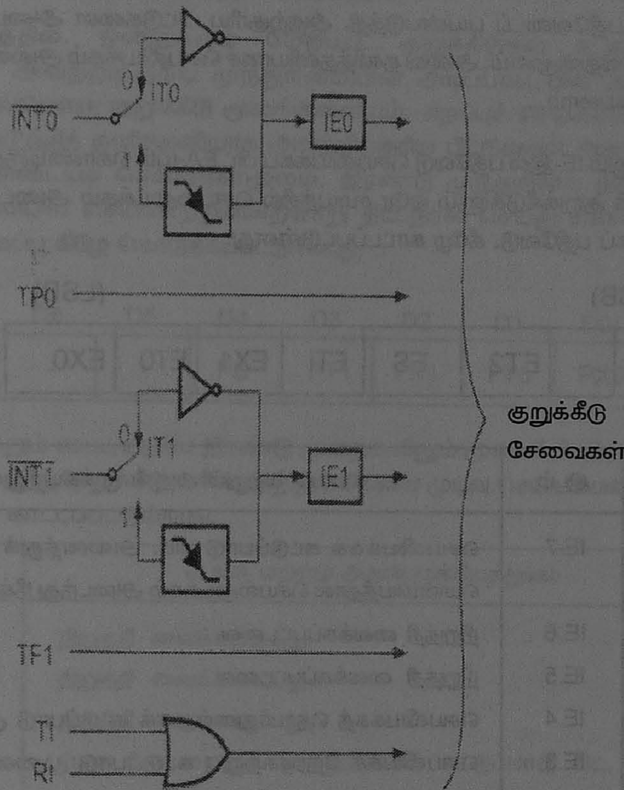
(3) புறக் குறுக்கீடு 1 (IE1)

(4) நேரச்சுற்றுக் குறுக்கீடு 1 (TF1)

(5) தொடர் துறைமுகக் குறுக்கீடு (TIFR1)

இந்தக் குறுக்கீடுகள் அனைத்தும், படம் 5.21-இல் காட்டப்பட்டுள்ளன. $\overline{INT0}$ மற்றும் $\overline{INT1}$ ஆகிய புறக் குறுக்கீடுகள், TCON பதிவேட்டிலுள்ள IT0 மற்றும் IT1 பிட்டுகளைப் பொறுத்து, ஒவ்வொன்றும் மட்ட அல்லது மாறும் செயலியக்கமுடையதாக இருக்கலாம். TCON பதிவேட்டிலுள்ள முனைகள் IE0 மற்றும் IE1, இந்தக் குறுக்கீடுகளை உண்மையாக உருவாக்குகின்ற கொடிகளாகும். ஒரு புறக் குறுக்கீடு உருவாகியவுடன், கட்டுப்பாடு அதற்குரிய சேவை துணை நிரற்செயன்முறைக்குச் சென்றவுடன், அந்தக் குறுக்கீடு மாற்று முறையில் செயலாற்றுவதாகத் தானாகவே காலி (0) யாகி விடும். மாறாக, அது மட்டம் முறையில் செயலாற்றுவதாக இருந்தால், அதற்குரிய குறுக்கீடு கொடி தானாகவே காலியாகாது. அதனை மென்மியம் முறையில்தான் காலி செய்யவேண்டும்.

நேரச்சுற்று 0 மற்றும் நேரச்சுற்று 1 ஆகியவற்றின் குறுக்கீடுகள், TF0 மற்றும் TF1 என்கிற கொடிகளால் உருவாகின்றன. இந்தக் கொடிகள், அதற்குரிய நேரச்சுற்றின் செயல் முழுவதுமாக முடிந்த பின்பு தாமாகவே அமைவு நிலையை அடையும். நேரச்சுற்றுக் குறுக்கீடு உருவாகியவுடன், நுண்கட்டுப்படுத்தியானது குறிப்பிட்ட சேவை துணைநிரற் செயன்முறை (Service sybroutine) க்குச் சென்று, அதனை செயல்படுத்தத் தொடங்கிய உடனேயே அதற்குரிய குறுக்கீடு கொடி தானாகவே காலியாகிவிடும்.



படம் 5.21 MCS - 51 குறுக்கீடு அமைப்பு

தொடர் துறைமுகக் குறுக்கீடு (Serial Port Interrupt) SCON பதிவேட்டிலுள்ள TI மற்றும் RI என்கிற பிட்டுகளின் வாதியல் OR செயலின்மூலம் உருவாகின்றது. துணை நிரற்செயன்முறைக்குச் செல்லும்போது, இந்தக் கொடிகள் இரண்டுமே வன்மியத்தால் நீக்கப்படுவதில்லை. தொடர் துறைமுகக் குறுக்கீடு உருவாகிய பின்பு அந்தக் குறுக்கீடு TI-க்குரியதா அல்லது RI-க்குரியதா என்பதைச் சேவை நிரறசெயன்முறைக் கண்டுபிடிக்கும். பின் மென்மியத்தால் அந்தப் பிட் நீக்கப்படும். அனுப்புச் செயல் முடிந்தவுடன், TI கொடி அமைவு நிலையை அடையும். அதே போன்று ஏற்புச் செயல் முடிந்தவுடன், RI கொடி அமைவு நிலையை அடையும்.

குறுக்கீடுகளை உருவாக்கிய அனைத்துப் பிட்டுகளையும், மென்மியம் முறையின்மூலம் அமைவு நிலை அல்லது நீக்கக் காலி நிலையை அடையச் செய்யும். இவ்வாறு உருவாக்கப்படுகின்ற குறுக்கீடுகள், சாதாரணக் குறுக்கீடுகள் போன்றே செயல்படும். அதாவது மென்மியம் முறையில் குறுக்கீடுகளை உருவாக்கியும் கொள்ளலாம் மற்றும் முடிவு பெறாத குறுக்கீடுகளை நீக்கியும் கொள்ளலாம்.

இந்த மாதிரியான ஒவ்வொரு குறுக்கீடு மூலங்களையும், IE என்கிற சிறப்புச் செயல்சார் பதிவேடைப் பயன்படுத்தி, அதற்குரிய பிட்டுகளை அமைவுநிலை அல்லது நீக்கம் செய்வதன்மூலம், அவை தனித்தனியாகச் செயலியக்கம் அல்லது செயல்முடக்கம் பெறச் செய்யலாம்.

மேலும் IE-இல் பதிவேடு செயலியக்கப் பிட் EA-யும் கொண்டிருக்கும். EA=0 எனில், அனைத்துக் குறுக்கீடுகளும் ஒரே சமயத்தில் செயல்முடக்கம் அடையும். IE - குறுக்கீடு செயலியக்கப் பதிவேடு, கீழே காட்டப்பட்டுள்ளது.

(MSB) (LSB)

EA	-	ET2	ES	ETI	EX1	ET0	EX0

குறி	இடம்	பெயர் மற்றும் அதன் முக்கியத்துவம்
EA	IE.7	செயலியக்கக் கட்டுப்பாடு பிட், அனைத்துக் குறுக்கீடுகளும் மென்மியத்தால் செயல்முடக்கம் அடைந்து நீக்கப் பயன்படும்.
	IE.6	நிறுத்தி வைக்கப்பட்டவை.
ET2	IE.5	நிறுத்தி வைக்கப்பட்டவை.
ES	IE.4	செயலியக்கத் தொடர்துறைமுகக் கட்டுப்பாடு முனை.
ET1	IE.3	செயலியக்க நேரச்சுற்று 1 கட்டுப்பாடு முனை. நேரச்சுற்று / எண்ணி 1 –லிருந்து குறுக்கீடுகளை மென்மியத்தால் செயலியக்கம் / செயல் முடக்கம் அடையச் செய்யப் பயன்படுகிறது. அமைவு / நீக்கம் மென்மியத்தால் நடைபெறும்.
EX1	IE.2	செயலியக்கப் புறக் குறுக்கீடு 1 கட்டுப்பாடு முனை அமைவு / நீக்கம். இது INT1-லிருந்து குறுக்கீடுகளை மென்மியத்தால் செயலியக்கம் / செயல்முடக்கம் பெறப்பயன்படும்.
ET0	IE.1	செயலியக்க நேரச்சுற்று 0 கட்டுப்பாடு முனை. நேரச்சுற்று / எண்ணி 0லிருந்து குறுக்கீடுகளை மென்மியத்தால் செயலியக்கம் / செயல்முடக்கம் அடையச் செய்யப் பயன்படுகிறது. அமைவு / நீக்கம் மென்மியத்தால் நடைபெறும்.
EX0	IE.0	செயலியக்கப் புறக் குறுக்கீடு 0 கட்டுப்பாடு முனை. அமைவு / நீக்கம். இது INT1-லிருந்து குறுக்கீடுகளை மென்மியத்தால் செயலியக்கம் / செயல்முடக்கம் பெறப் பயன்படும்.

முந்துரிமை மட்ட அமைப்பு : (Priority Level Structure)

ஒரே நேரத்தில், ஒன்றுக்கும் மேற்பட்ட குறுக்கீடுகள் உருவானால், நுண்கட்டுப்படுத்தி அவற்றிற்குரிய முந்துரிமையின் அடிப்படையில் அவற்றைச் செயல்படுத்தும். ஒவ்வொரு குறுக்கீடு மூலங்களையும், சிறப்புச் செயல்சார் பதிவேடு (SFR) IP-ஐ பயன்படுத்தித் தனித்தனியாக அவற்றிற்குரிய பிட்டுகளை அமைவுநிலை அல்லது நீக்கம் அடையச் செய்வதன்மூலம், இரண்டு முந்துரிமை நிலைகளில் ஏதாவதொரு நிலையில் செயலாற்றுகின்றவாறு திட்டநிரல் செய்து கொள்ளலாம். IP பதிவேடின் அமைப்பு கீழே கொடுக்கப்பட்டுள்ளது.

(MSB) D7	D6	D5	D4	D3	D2	D1	D0 (LSB)
-	-	-	PS	PT1	PX1	PT0	PX0

IP பதிவேடு, பிட் மற்றும் பைட் ஆகிய இரண்டு முறைகளிலும் முகவரியிடத் தகுந்த ஒரு 8 பிட் பதிவேடாகும். இது குறுக்கீடுகளின் முந்துரிமையை முடிவு செய்யபயன்படுகிறது. இதன் விவரம் கீழே காட்டப்பட்டுள்ளது.

குறி	இடம்	பெயர் மற்றும் அதன் முக்கியத்துவம்
-	IP.7	நிறுத்தி வைக்கப்பட்டது.
-	IP.6	நிறுத்தி வைக்கப்பட்டது.
-	IP.5	நிறுத்தி வைக்கப்பட்டது.
PS	IP.4	தொடர் துறைமுக முந்துரிமைக் கட்டுப்பாடு பிட், தொடர் துறைமுகக் குறுக்கீட்டின் முந்துரிமை நிலையை நிர்ணயிக்கப் பயன்படுகிறது. தொடர் துறைமுகத்திற்காக உயர் / தாழ் முந்துரிமைக் குறுக்கீடுகளை மென்மியத்தால் நிர்ணயிக்கலாம். அமைவு / நீக்கம் மென்மியத்தால் நடைபெறும்.
PT1	IP.3	நேரச்சுற்று 1 முந்துரிமைக் கட்டுப்பாடு முனை. நேரச்சுற்று 1இன் குறுக்கீடு, முந்துரிமை நிலையை நிர்ணயிக்கப் பயன்படுகிறது. PT 1 = 1 எனில், உயர் முந்துரிமையுடன் செயலாற்றும். நேரச்சுற்று / எண்ணி 1க்கான உயர் / தாழ் முந்துரிமைக் குறுக்கீடுகளை, மென்மியத்தால் நிர்ணயம் செய்யலாம். அமைவு / நீக்கம் மென்மியத்தால் நடைபெறும்.
PX1	IP.2	புறக் குறுக்கீடு 1, முந்துரிமைக் கட்டுப்பாடு முனை. புறக் குறுக்கீட்டு 1ன் முந்துரிமையை நிர்ணயம் செய்யப் பயன்படுகிறது. INT1-லிருந்து உயர் / தாழ் முந்துரிமைக் குறுக்கீடுகளை மென்மியத்தால் நிர்ணயிக்கலாம். அமைவு / நீக்கம் மென்மியத்தால் நடைபெறும்.

குறி	இடம்	பெயர் மற்றும் அதன் முக்கியத்துவம்
PT0	IP. 1	நேரச்சுற்று 0 முந்துரிமை கட்டுப்பாடு முனை. நேரச்சுற்று 0இன் குறுக்கீட்டின் முந்துரிமையை நிர்ணயம் செய்யப் பயன்படுகிறது. நேரச்சுற்று / எண்ணி 0க்கான உயர் / தாழ் முந்துரிமைக் குறுக்கீடுகளை மென்மியத்தால் நிர்ணயம் செய்யலாம். அமைவு/நீக்கம் மென்மியத்தால் நடைபெறும்.
PX0	IP.0	புறக் குறுக்கீடு 0 முந்துரிமைக் கட்டுப்பாடு முனை. புற குறுக்கீடு 0இன் முந்துரிமையை நிர்ணயம் செய்யப் பயன்படுகிறது. INT0- விருந்து உயர் / தாழ் முந்துரிமைக் குறுக்கீடுகளை மென்மியத்தால் நிர்ணயிக்கலாம். அமைவு/ நீக்கம் மென்மியத்தால் நடைபெறும்.

தாழ்ந்த முந்துரிமை கொண்ட ஒரு குறுக்கீடு, உயர்ந்த முந்துரிமை கொண்ட வேறொரு குறுக்கீட்டினால் குறுக்கீடு செய்யப்படலாம். ஆனால், வேறொரு தாழ்ந்த முந்துரிமை கொண்ட குறுக்கீட்டினால் குறுக்கீடு செய்ய முடியாது. ஒரு உயர்ந்த முந்துரிமை கொண்ட குறுக்கீடை, வேறு எந்த குறுக்கீடும் குறுக்கீடு செய்ய முடியாது.

வெவ்வேறு முந்துரிமை நிலைகள் கொண்ட இரண்டு குறுக்கீடுகள் ஒரே நேரத்தில் ஏற்றுக்கொள்ளப்பட்டால், முதலில் உயர்ந்த முந்துரிமை கொண்ட குறுக்கீட்டின் வேண்டுகோளை ஏற்று, அதனைச் செயல்படுத்தும். ஒரே முந்துரிமை நிலைகள் கொண்ட இரண்டு குறுக்கீடுகள் ஒரே நேரத்தில் ஏற்றுக்கொள்ளப்பட்டால், எந்தக் குறுக்கீடின் வேண்டுகோளை ஏற்கவேண்டும் என்பது, உள்அகக் குறுக்கீடு வரிசைகளின்மூலம் கண்டுப்பிடிக்கப்படும். அதாவது, ஒவ்வொரு முந்துரிமைக்கு உள்ளாகவும், இரண்டாவது முந்துரிமை அமைப்பு இருக்கும். இது கணக்கீடு வரிசை செய்வதன் மூலம் கண்டுபிடிக்கப்படும். முந்துரிமை நிலைக்கு உள்ளாக இருக்கின்ற இரண்டாவது முந்துரிமை எவ்வாறு இருக்கும் என்பது, கீழே உள்ள அட்டவணையில் காட்டப்பட்டுள்ளது.

வ. எண்	குறுக்கீடு	வெக்டார் இருப்பிடம்	முந்துரிமை நிலைகள்
1.	புற வன்மியம் குறுக்கீடு 0 (INT0)	0003 H	(உயர்ந்த)
2.	நேரச்சுற்று 0 குறுக்கீடு (TF0)	000B H	
3.	புற வன்மியம் குறுக்கீடு 1 (INT1)	0013 H	
4.	நேரச்சுற்று 1 குறுக்கீடு (TF1)	001 B H	
5.	தொடர் அனுப்புக் குறுக்கீடு RI மற்றும் TI	00 23 H	(தாழ்ந்த)

புறக் குறுக்கீடுகள் :

8051 நுண்கட்டுப்படுத்தி, இரண்டு புறக் குறுக்கீடுகளைக் கொண்டுள்ளது. புற குறுக்கீடுகளைத் தேவைக்கேற்ப ஊக்கப்பட்ட நிலை அல்லது ஊக்கப்பட்ட பெயர்ச்சி ஆகிய முறைகளில் செயல்படுத்திக் கொள்ளலாம். TCON பதிவேடிலுள்ள IT2 மற்றும் IT0 பிட்டுகளைப் பயன்படுத்தித் தேவையான நிலையைச் செயல்படுத்தலாம். இதில் IT1 பிட் புறக் குறுக்கீடு 1க்காகவும் மற்றும் IT0 பிட் புறக் குறுக்கீடு 0-ற்காகவும் பயன்படுத்தப்படுகின்றன. ITx=0 எனில், புறக் குறுக்கீடு x அதற்குரிய INTx முனையில் தாழ்நிலை கொண்ட சைகை வந்தால் சரியான தொடக்கியாக ஏற்றுக்கொள்ளும். மாறாக, ITx=1 எனில், புறக் குறுக்கீடு x அதற்குரிய INTx முனையில் விளிம்புத் தொடக்கி சைகை வந்தால் மட்டுமே சரியான குறுக்கீடு என ஏற்றுக்கொள்ளும். இந்த முறையில், INTx முனையில் - அடுத்தடுத்த மாதிரிகள் ஒரு சுழற்சியில் உயர்நிலைக்கும் அடுத்த சுழற்சியில் தாழ்நிலைக்கும் சென்றால், TCON-இல் உள்ள குறுக்கீடு வேண்டுகோள் கொடி IEx அமைவுநிலையில் இருக்கும். கொடி பிட் IEx குறுக்கீடின் வேண்டுகோளை ஏற்கும்.

ஒவ்வொரு எந்திரச் சுழற்சியின்போதும், புறக் குறுக்கீடு முனைகள் ஒன்று தேர்வு செய்யப்பட்டுக் குறைந்தது 12 அலையியற்றி நேரங்களுக்கு உயர் அல்லது தாழ் உள்ளீடை வைத்திருக்க வேண்டும். இதன் பின்னர்தான், சரியான முனையின் தேர்வு தீர்மானிக்கப்படும். புறக் குறுக்கீடு ஊக்கப்பட்ட பெயர்ச்சி எனில், குறைந்தது ஒரு எந்திரச் சுழற்சிக்குப் புறக் குறுக்கீடு முனை உயர்நிலையில் இருக்க வேண்டும். அதன் பிறகு குறைந்தது ஒரு எந்திரச் சுழற்சிக்குத் தாழ்நிலையில் இருக்க வேண்டும். அப்போதுதான் பெயர்ச்சி ஏற்படுவதாகத் தீர்மானிக்கப்பட்டு IEx என்கிற குறுக்கீடு கொடி அமைவு நிலையை அடையும். குறுக்கீடு ஏற்பட்டுக் கட்டுப்பாடு அதற்குரிய சேவை துணைநிரற் செயன்முறைக்குச் சென்றவுடன்,க் CPU வினால் IEx தன்னிச்சையாகவே நீக்கப்படும்.

புறக் குறுக்கீடு ஊக்கப்பட்ட - நிலையில் செயல்படுவதாயிருந்தால், உண்மையான குறுக்கீடு உருவாகும் வரையிலும் புறக்குறுக்கீடு அந்த வேண்டுகோளை செயலியக்க நிலையில் வைத்திருக்க வேண்டும். அதாவது அந்தக் குறுக்கீடு ஏற்றுக்கொள்ளப்படுகின்ற வரை, அதற்குரிய குறுக்கீடு முனை தாழ்நிலையில் இருக்கவேண்டும். அதற்குரிய திரன்நிரன்செயன்முறை முடிவடைவதற்கு முன்பாக, அதற்குரிய குறுக்கீடு கொடியை மென்மியம் முறையில் நீக்கவேண்டும்.

5.17. தொடர் துறைமுகமும் அதன் திட்டநிரலிடுதலும் : (Serial port and their programming)

8051 நுண்கட்டுப்படுத்தியிலுள்ள தொடர் துறைமுகம், இருவழியில் செயல்படக்கூடியது. அதாவது, ஒரே நேரத்தில் பரப்பு மற்றும் ஏற்புச் செயல்களைச் செய்யும் தன்மையைப் பெற்றிருக்கும். இதில் பதிவேடு, தரவினை வைத்திருப்பதற்காகப் பயன்படுகிறது. தரவு அனுப்புதலைப் பதிவேடு கட்டுப்படுத்தும். தரவு வேகத்தைப் பதிவேடு SCON கட்டுப்படுத்தும். தரவு வேகத்தைப் பதிவேடு PCON கட்டுப்படுத்தும் மற்றும் முனை RXD (P3.0) மற்றும் TXD (3.1) தரவு மாற்றும் செயலைச் செய்யும். தொடர் துறைமுக ஏற்பு மற்றும் பரப்புப் பதிவேடுகள் இரண்டையும் SBUF என்கிற சிறப்புச் செயல்சார் பதிவேடுகளின்மூலம் பயன்படுத்த முடியும். SBUF-இல் எழுதுவது பரப்புப் பதிவேடைச் சுமையேற்றுவதாகும். தனியான ஏற்புப் பதிவேடைப் பயன்படுத்தி SBUF-வைப் படிக்கலாம். 8051இன் தொடர் துறைமுகம் ஒரு ஏற்பு – தாங்கியாகும். அதாவது, ஏற்கனவே ஏற்கப்பட்ட பைட்டை ஏற்புப் பதிவேடிலிருந்து படிப்பதற்கு முன்பாக, இரண்டு பைட்டை ஏற்க ஆரம்பித்து விடும்.

SBUF பதிவேடு, உண்மையில் இரண்டு பதிவேடுகளைக் கொண்டிருக்கும். அதில் TXD முனை வழியாகப் பரப்பக்கூடிய தரவைப் பரப்புப் பதிவேடில் எழுதலாம். RXD முனை வழியாக ஏற்கப்படுகின்ற தரவானது, ஏற்புப் பதிவேடிலிருக்கும். இதனைத் தேவைக்கேற்பப் பயன்படுத்திக் கொள்ளலாம்.

8051 தொடர் தரவு, அனுப்புதலுக்காக நான்கு திட்டமிடத்தக்க முறைகளைக் கொண்டுள்ளது. SCON பதிவேட்டிலுள்ள SM0 மற்றும் SM1 முனைகளை அமைத்து (1) ஒரு குறிப்பிட்ட முறையைத் தேர்வு செய்யலாம். SCON மற்றும் BCON பிட்டின் அமைப்புகள், படம் 5.22 (அ) மற்றும் (ஆ) களில் காட்டப்பட்டுள்ளன.

(MSB) 7	6	5	4	3	2	1	0 (LSB)
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

குறி	இடம்	பெயர் மற்றும் அதன் முக்கியத்துவம்
SM0	SCON.7	தொடர் துறைமுக முறைக் கட்டுப்பாடு முனை 0 . மென்மியத்தால் அமைவு / நீக்க நிலையை அடையப் பயன்படுகிறது.
SM1	SCON.6	தொடர் துறைமுக முறைக் கட்டுப்பாடு முனை 1. மென்மியத்தால் அமைவு / நீக்க நிலையை அடையப் பயன்படுகிறது.
SM2	SCON.5	தொடர்துறைமுகக் கட்டுப்பாடு முனை 2. (மென்மியத்தால் அமைவு நிலையில் வைக்கப்படும்)
REN	SCON.4	ஏற்பி செயலியக்கக் கட்டுப்பாடு பிட் இது தொடர் வரிசைத் தரவு உள்வாங்கலைச் செயலியக்கமடையச் செய்கின்ற முனை ஆகும். இதனை மென்மியம் மூலம் அமைவு / நீக்க நிலையாக அமைத்துக்கொள்ளலாம்.
TB8	SCON.3	பரப்பு பிட் UART முறையில் செயலாற்றுகின்றபோது அனுப்ப வேண்டிய 9-ஆவது பிட்டை இது கொண்டிருக்கும். மென்மிய முறையில், இதனை அமைவு / நீக்க நிலையில் வைக்கலாம்.
RB8	SCON.2	ஏற்பு பிட் 8. முறை 2 மற்றும் 3களில் செயலாற்றுகின்றபோது ஏற்கப்பட்ட 9-வது பிட், இதனைக் குறிக்கிறது.
T1	SCON.1	பரப்பு க்குறுக்கீடு கொடி. பைட்டானது அனுப்பப்பட்டவுடன், வன்மிய முறையால் அமைவு நிலையை அடையும். சேவை முடிந்த பின்பு மென்மியம் முறையில் நீக்க முடியும்.
R1	SCON.0	ஏற்புக் குறுக்கீடு கொடி. பைட் ஏற்றவுடன் வன்மிய முறையால் அமைவு நிலையை அடையும். சேவை முடிந்தபின்பு மென்மியம் முறையில் மட்டும் இதனை நீக்க முடியும்.

குறிப்பு : SM0, SM1 கொண்டு நிலையைத் தேர்வு செய்தல்.

முறை	SM0	SM1	
0	0	0	பெயரும் பதிவேடு பாட் = f/12 (நகர்வுப் பதிவேடு)
1	0	1	8 பிட் UART மாறுகிற தரவு வேகம்.
2	1	0	9 பிட் UART, நிலையான தரவு வேகம். பாட் = f/32 அல்லது 4/64
3	1	1	9 பிட் UART, மாறுகிற தரவு வேகம்.

படம் 5.22 (அ) SCON தொடர்வரிசை துறைமுகக் கட்டுப்பாடு / நிலைப்பாட்டுப் பதிவேடு.

(MSB) 7	6	5	4	3	2	1	0 (LSB)
SM0D	-	-	-	GF1	GF0	PD	IDL

குறி	இடம்	பெயர் மற்றும் அதன் முக்கியத்துவம்
SMOD	PCON.7	தொடர் வரிசை பாட் விகிதம் மாற்று பிட். திட்டநிரலால் 1 என அமைக்கப் பாட் விகிதம், இரண்டு பங்காக இருக்கும்.
	PCON.6.4	வரையறுக்கப்படாதவை.
GF1	PCON:3	பொது நோக்கப் பயனாளர் கொடி பிட் 1. திட்டநிரலால் அமைவு / நீக்க நிலைக்குக் கொண்டு வர முடியும்.
GF0	PCON.2	பொதுநோக்கப் பயனாளர் கொடி பிட் O. திட்டநிரலால் அமைவு / நீக்க நிலையை அடைய முடியும்.
PD	PCON.1	திறன் கீழ் பிட். இந்தப் பிட் அமைவு நிலையிலிருந்தால், திறன் கீழ் முறையில் செயலாற்றும்.
IDL	PCON.0	முறை பிட். இந்தப் பிட் அமைவு நிலையிலிருந்தால், முறையில் செயலாற்றும்.

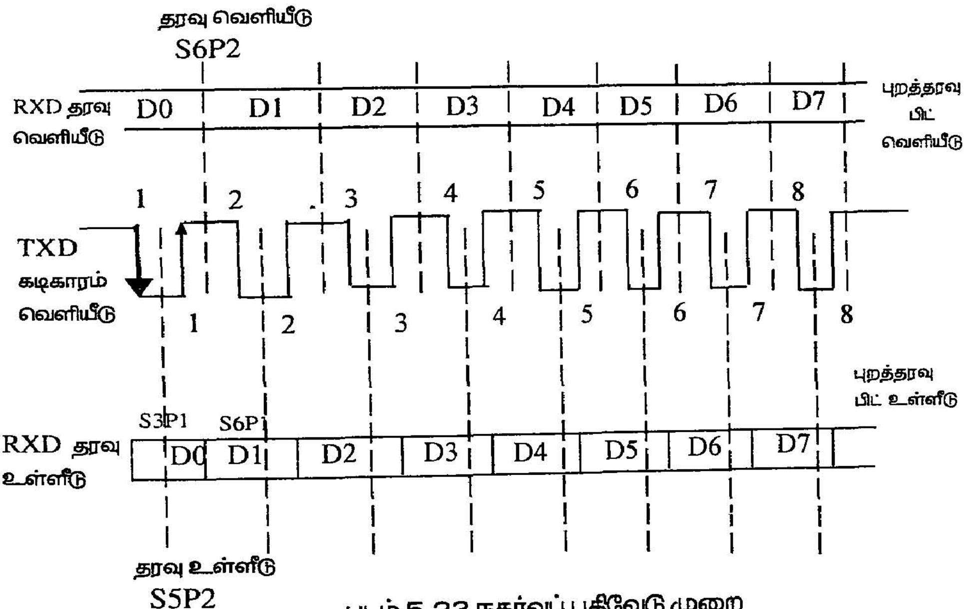
படம் 5.22 (ஆ) PCON பிட் அமைப்பு

தொடர் வரிசைத் துறைமுகத்திற்கான செயற்பாட்டு முறைகள் :
(Operating Modes for serial port)

தொடர் வரிசைத் தரவு முறை 0 : நகர்வுப் பதிவேடு முறை

இந்த முறையில் RXD-இன் வழியாக மட்டுமே தொடர் வரிசை உட்படும். தரவு மற்றும் வெளியே செல்லும். TXD ஆனது நகர்வுக் கடிகையை வெளியிடும். இந்த முறையில் 8 பிட்டுகள் அனுப்பப்படும் அல்லது ஏற்கப்படும். இதில் LSB பிட் முதலில் அனுப்பப்படும் அல்லது ஏற்கப்படும். இதில் பாட் விகிதம் ½ நிலையாக இருக்கும்.

நகர்வுப் பதிவேடு முறை படம் 5.23-இல் காட்டப்பட்டுள்ளது.



படம் 5.23 நகர்வுப் பதிவேடு முறை

இந்த முறையில் செலுத்தீடு செய்யும்போது, RXD-லிருந்து தரவானது வெளியேற்றப்படும். S6P2-இன் இறங்கு விளிம்பின்போது, தரவானது மாற்று அல்லது ஒரு கடிகாரத்துடிப்பிற்குப் பிறகு நகர்வுக் கடிகை TXD-இன் வெளியீடின் ஏறு விளிம்பின்போது மாறும். S6P2 என்ற நிலையில் ஏற்படுகின்ற **"SBUF-இல் எழுது"** என்கிற சைகையின்போது, நகர்வுப் பதிவேடின் 9-வது பிட் இடத்தில் 1ஐ சுமையேற்றும் மற்றும் TXDயிடம் செலுத்தீடு செயலைச் செய்யும்படி கூறுகின்றது.

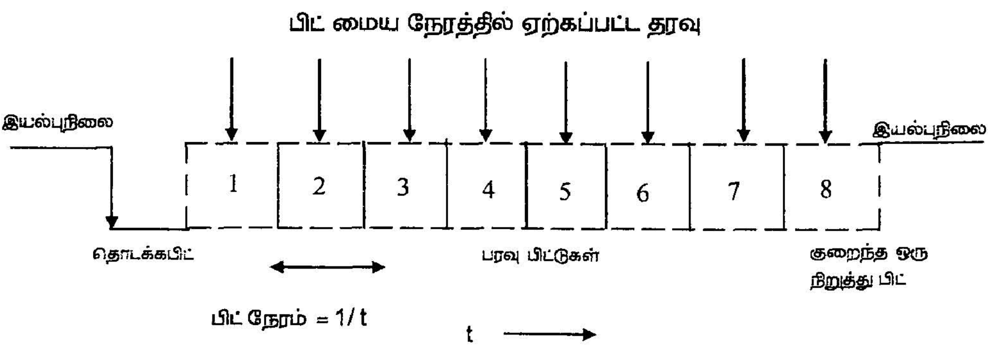
"SBUF-இல் எழுது" என்கிற சைகை உருவாகி, "அனுப்பு" (SEND) சைகை உருவாவதற்கு முன்பாக, அதற்கு இடைப்பட்ட கால இடைவெளியில் ஒரு முழு எந்திரச் சுழற்சி வீணாகிவிடும். நகர்வுக் கடிகை ஒவ்வொரு எந்திரச் சுழற்சியின் S3, S4 மற்றும் S5 நிலைகளின்போது தாழ்நிலையிலும் மற்றும் S6, S1 மற்றும் S2 நிலைகளின் போது உயர்நிலைகளிலும் இருக்கும். ஒவ்வொரு எந்திரச் சுழற்சியின் S6P2 நிலையின்போதும் அனுப்புச் சைகை செயலியக்க நிலையில் இருக்கும். அனுப்புச் சைகை செயலியக்கம் அடையும்போது, நகர்வுப் பதிவேட்டிலுள்ள மதிப்பானது ஒரு பிட் வலது புறமாக நகரும். "SBUF-இல் எழுது" சைகை உருவாகி, அதனைத் தொடர்ந்து வருகின்ற பத்தாவது எந்திரச் சுழற்சியின் S1P1நிலையின்போது, அனுப்புச் சைகை விலக்கப்பட்டுவிடும் மற்றும் T1 அமைவு நிலைக்குச் செல்லும்.

RXD-னுள்ளே ஏற்கப்பட்ட தரவு TXD -இல் உருவாகின்ற நகர்வுக் கடிகையுடன் ஒத்திருக்க வேண்டும். S5P2 -இன் இறங்கு விளிம்பின்போது தரவு, தேர்வு செய்யப்பட்டு, நகர்வுக் கடிகையின் ஏறுகின்ற விளிம்பின்போது SBUF-ற்கு நகர்த்தப்படும்.

கணிப்பொறிகளுக்கிடையே தரவினை அனுப்புவதற்கு முறை 0 உகந்ததல்ல. ஆனால், அதிவேகத் தொடர் வரிசை, தரவுச் சேமிப்பு முறையின்படி தனி வாதியியல் சுற்றைப் பயன்படுத்தித் தரவின் வேகத்தை அதிகரிக்க முடியும். முறை 0-இல் பயன்படுத்தப்படும் பாட் விகிதம், திட்ட அளவைவிட அதிகமாக இருக்கும். உதாரணமாக, 6 மெகா ஹர்ட்ஸ் படிகத்திற்கான நகர்வு வேகம், 500 கிலோ ஹர்ட்ஸ் ஆகும்.

தொடர் வரிசைத் தரவு முறை 1 (8 பிட் UART) :

இந்த முறையில், 10 பிட்டுகள் TXD -இன் வழியாக அனுப்பப்படும் அல்லது TXD -இன் வழியாக ஏற்கப்படும். SM0 மற்றும் SM1 ஆகியவை 01 எனில், SBUF முழு இரட்டை ஏற்பி / பரப்பியாக செயல்படும். அப்போது ஒரே நேரத்தில் தரவினை அனுப்ப மற்றும் ஏற்க முடியும். படம் 5.24-இல் தரவுச் சொல்லின் உருவமைப்பு காட்டப்பட்டுள்ளது.



படம் 5.24 தரவுச் சொல்லின் உருவமைப்பு

அனுப்பப்படும் தரவானது ஆரம்ப பிட், எட்டுத் தரவு பிட்டுகள் (LSB முதலில்) மற்றும் ஒரு நிறுத்துப் பிட் ஆகிய பிட்டுகளாக அனுப்பப்படும். அனைத்துப் பிட்டுகளும் அனுப்பப்பட்டவுடன், குறுக்கீடு கொடி TI அமைவு நிலைக்குச் செல்லும். ஒவ்வொரு பிட்டின் இடைவெளியும் பாட் விகிதம் அதிர்வெண்ணிற்கு நேர்மாறாக இருக்கும். மேலும் அந்த இடைவெளி முழுவதும், ஒவ்வொரு பிட்டும் உயர் அல்லது தாழ் நிலையைப் பேணும் நிலையிலேயே இருக்கும்.

அதே வரிசையில் ஏற்கப்படுகின்ற தரவும் கிடைக்கும். ஆரம்பப் பிட்டின் இறங்கு விளிம்பால் ஏற்புச் செயல் தொடங்கப்படுகிறது. பெறப்பட்ட முதல் பிட்டான தொடக்க பிட் '0' வாக இருந்தால் மட்டுமே, ஏற்புச் செயல் தொடர்ந்து நடைபெறும். பெறப்பட்ட தரவு SBUF பதிவேட்டிற்குச் செல்லும் மற்றும் நிறுத்துப் பிட் SCON-லுள்ள RB8 என்கிற பிட்டிற்குச் செல்லும் மற்றும் RI அமைவு நிலையை அடையும்.

தரவு பிட்டுகள் ஏற்புப் பதிவேட்டிற்குத் திட்டமிடத்தக்க பாட், பட்டை விகிதத்தில் நகரும். RI=0 மற்றும் முறை பிட் SM2=0 அல்லது நிறுத்துப் பிட் =1 என இருந்தால் மட்டுமே பெறப்பட்ட தரவு SBUF மற்றும் RBS-இல் சுமக்கப்படும். RI=0 என்பது, திட்டநிரல் முந்தைய தரவு பைட்டைப் படித்ததைக் குறிக்கும் மற்றும் அடுத்த தரவினைப் பெறுவதற்குத் தயாராக இருக்கும். SM2-இன் நிலை எதுவாக இருந்தாலும், தரவினை SBUF-க்கு மாற்றும் நிறுத்துப் பிட்டின் செயல் முழுமை அடையும். *SM2=0 என்பது நிறுத்துப் பிட் நிலையுடன்* பெறப்பட்ட பைட்டைச் செயலியக்கம் அடையச் செய்வதாகும். இந்த முறையைப் பயன்படுத்துவதற்கான நிபந்தனை வரம்பிற்குட்படுத்தப்பட்டது. ஆனால், முறை 2 மற்றும் முறை 3 களில் மிகவும் பயனுள்ளதாகும்.

ஆரம்பப் பத்து பிட்டுகளில், தொடக்கப் பிட்டை விலக்கி, எட்டுத் தரவு பிட்டுகளும் SBUF-க்குச் செல்லும். SCON-இன் RB8-இல் நிறுத்துப் பிட் சேமிக்கப்படும். RI=1 என்பது, புதிய தரவு பைட் பெறப்பட்டுள்ளதைக் குறிக்கும்.

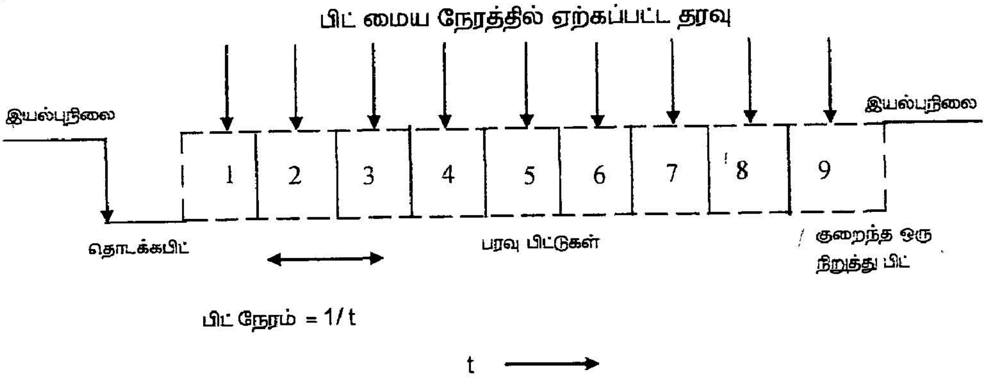
முறை 1 பாட் விகிதங்கள் : (Mode 1 Baud Rate)

நேரச்சுற்று 1-ஐ பயன்படுத்தி, முறை 1-க்கான பாட் விகிதத்தை உருவாக்கலாம். பாட் அதிர்வெண்ணைக் கணக்கிட நேரச்சுற்றின் பொங்கி வழி கொடி பயன்படுகிறது.

$$f_{baud} = \frac{2^{SMOD}}{32} \times \text{(நேரச்சுற்று 1-இன் பொங்கிவழி அதிர்வெண்)}$$

தொடர்வரிசைத் தரவு முறை 2 - பன்முறைப்படுத்தி முறை :
(Serial Data Mode 2 - Multiprocessor Mode)

முறை 2, முறை 1 போன்றிருக்கும். முறை 2-இல் செயலாற்றுகின்ற போது 11 பிட்டுகள் TXD -இன் வழியாக அனுப்பப்படும் அல்லது RXD வழியாகப் பெறப்படும். இது ஒரு தொடக்க பிட் (0) 8 தரவு பிட்டுகள் (LSB முதலில்). திட்டமிடத்தக்க 9-ஆவது தரவு பிட் மற்றும் ஒரு நிறுத்துப் பிட் (1) என மொத்தம் 11 பிட்டுகளைக் கொண்டிருக்கும். இதன் தரவு அமைப்பானது, படம் 5.25-இல் காட்டப்பட்டுள்ளது.



படம் 5.23 பன்முறைப்படுத்தித் தரவுச் சொல்

அனுப்புச் செயலின்போது, 9-ஆவது தரவு பிட்டை (SCON-லுள்ள TB8) 0 அல்லது 1 என வைத்துக்கொள்ளலாம். தரவினைப் பெறும்போது 9-ஆவது பிட் SCON -லுள்ள RB8-க்குச் செல்லும். இந்த முறைக்கான பாட் விகிதத்தை $f_{osc}/32$ அல்லது $f_{osc}/64$ எனத் தேர்வு செய்து கொள்ளலாம்.

அனுப்புச் செயலின்போது, SBUF என்கிற பதிவேட்டைச் சேருமிடப் பதிவேடு (destination register) ஆகக் கொண்டு கட்டளையின்மூலம் ஆரம்பிக்கப்படுகின்றது.

$$f_{band^2} = \frac{2^{SMCD}}{64d} \times \text{அலையியற்றி அதிர்வெண்}$$

முறை 0 வைப் போல், முறை 3இல் பாட் விகிதம் மிக அதிகம். இந்த அதிகத் தரவு விகிதம், பன்முறைப்படுத்தியின் அதிகமான பயன்பாடுகளுக்குத் தேவைப்படும். முறை 2க்கான RI-ஐ அமைக்கும் நிபந்தனை முறை, 1-ஐ போன்றதாகும். கடைசி பிட் பெறப்படும்போது RI=0வாக இருக்கவேண்டும். மற்றும் SM2-0 அல்லது பெறப்பட்ட ஒன்பதாவது பிட் 1 ஆக இருக்கவேண்டும். SM2=0 என்று இருந்தால் மட்டும், பெறப்பட்ட தரவினால் குறுக்கிடப்படும். அதனுடைய ஒன்பதாவது பிட் 0 என்றாகும். SM2 =1 எனில், தரவினுடைய 9-வது பிட் 0 என்றுள்ள செய்தியினால் குறுக்கிடப்படாது. மேற்கூறப்பட்டுள்ள ஏதாவதொரு தன்மை ஏற்படாமல் இருந்தால்கூடப் பெறப்பட்ட தகவல் முழுவதும் அழிந்து விடும். மற்றும் RI அமைவு நிலைக்குச் செல்லாது.

இந்தத் திட்டம் செய்தி அனுப்புகின்ற கணிப்பொறி, தேர்ந்தெடுத்த செய்தி பெறுகின்ற கணிப்பொறியுடன் மட்டும் சேயல்படும். மற்ற ஏற்புடைய கணிப்பொறிகளைக் குறுக்கீடு செய்யாமல் பேச அனுமதிக்கும் செய்தியினைப் பெறுகின்ற கணிப்பொறிகள், "பேசுபவர் "-லிருந்து "கவனமாகக் கேட்பவர்" அல்லது "செவிடாக உள்ளவர் " ஆகியவர்களால் குறியீடு செய்யப்பட்ட பைட்டுடன், ஒன்பதாவது தரவு பிட் 1 என அமைக்கக் கட்டளையிடும். ஒன்பதாவது பிட்டிலுள்ள 1 அனைத்து ஏற்பிகளையும் குறுக்கிடும். குறியீடு பைட்டுகளுக்கு பதில் கொடுப்பதற்கு ஏற்றாற்போல், அவைகளைத் திட்டமிட உத்தரவிடும். அதனுடைய முறையான SCON பதிவேடுகளில் SM2-இன் நிலையைத் திட்டநிரலிடும். தேர்ந்தெடுக்கப்பட்ட கேட்பவர்கள், 0 செய்திகளால் அமைக்கப்பட்ட ஒன்பதாவது பிட்டிற்குப் பதில் கொடுப்பார்கள்.

தொடர் வரிசை முறை 3 :

இந்த முறையில், 11 பிட்டுகள் TXD -இன் வழியாக அனுப்பப்படும் அல்லது RXD -இன் வழியாகப் பெறப்படும். இதில் தரவு பிட் ஒரு தொடக்க பிட் (0), 8 தரவு பிட்டுகள் (LSB முதலில்) ஒரு திட்டமிடத்தக்க 9வது தரவு பிட் மற்றும் ஒரு நிறுத்துப் பிட் (1) என மொத்தம் 11 பிட்டுகளைக் கொண்டிருக்கும். முறை 3-இன் செயல் முறை 2-ஐ போன்றே இருக்கும். ஆனால், பாட் விகிதத்தில் மட்டும் வேறுபடும். முறை 3-இன் பாட் விகிதம் மாறக்கூடியது.

நான்கு முறைகளிலும் செய்தி அனுப்புதலை, எந்தக் கட்டளையின் மூலமாகவும் ஆரம்பிக்கலாம். அவற்றில் SBUF பதிவேடு சேருமிடமாக பயன்படும். முறை 0-இல் செய்தி பெறப்படுவது REN = 1 மற்றும் RI=0 என்கிற நிபந்தனைகளைக் கொண்டு ஆரம்பிக்கப்படுகிறது. மற்ற முறைகளில் உள்ளே நுழையும் தரவின் தொடக்கப் (REN=1) பிட்டினால் ஏற்புடைய செயல் தொடங்கும். அட்டவணை 5.5-இல் நான்கு தொடர் வரிசை நிலைய முறைகள் தொகுக்கப்பட்டுள்ளன.

அட்டவணை 5.5 தொடர்வரிசை நிலைய முறைகள்

முறை	செலுத்தீட்டு அமைப்பு	பாட் விகிதம்
0	8 – தரவு பிட்டுகள்	1/12 அலையியற்றி அதிர்வெண்
1	10 – பிட் (தொடக்கப் பிட் + 8 தரவு பிட்டுகள் + நிறுத்துப் பிட்)	மாறக்கூடியது (மாறி)
2	11 – பிட் (தொடக்கப் பிட் + 8 தரவு பிட்டுகள் + திட்டமிடத்தக்க 9வது தரவு பிட் + நிறுத்துப் பிட்)	1/32 அல்லது 1/64 அலையியற்றி அதிர்வெண்
3	11 – பிட் (தொடக்கப் பிட் + 8 தரவு பிட்டுகள் + திட்டமிடத்தக்க 9வது தரவு பிட் + நிறுத்துப் பிட்)	மாறி

பாட் விகிதங்களை உருவாக்குதல் :

முறை 0-இல் தொடர்வரிசைத் துறைமுகம் :

$$\text{பாட் விகிதம்} = \frac{\text{அலையியற்றி அதிர்வெண்}}{12}$$

முறை 1-இல் தொடர்வரிசைத் துறைமுகம்

முறை 1இல் பாட் விகிதம் ஒரு மாறி. பாட் விகிதமானது நேரச்சுற்று 1 அல்லது நேரச்சுற்று 2 (8052-இல் மட்டும்) ஆல் உருவாக்க முடியும்.

$$\text{பாட் விகிதம்} = \frac{k \times \text{அலையியற்றி அதிர்வெண்}}{32 \times 12 \times [256 - TH1]}$$

SMOD = 0 எனில், k=1 ஆகும்

SMOD = 1 எனில், k=2 ஆகும் (SMOD என்பது PCON ஆகும். பதிவேடு ஆகும்)

TH 1-இன் மறுசுமையின் மதிப்பு

$$TH\,1 = 256 - \frac{k \times \text{அலையியற்றி அதிர்வெண்}}{384 \times \text{பாட் விகிதம்}}$$

TH 1-இன் மதிப்பு ஒரு முழு எண் ஆகும்.

முறை 2இல் தொடர்வரிசைத் துறைமுகம் :

இந்த முறையில் பாட் விகிதம் நிலையானது. PCON பதிவேட்டிலுள்ள SMOD பிட்டின் மதிப்பைப் பொறுத்தது.

பாட் விகிதம் = (1/32 அல்லது 1/64) x அலையியற்றி அதிர்வெண்

இந்த முறையில், நேரச்சுற்று எதுவும் பயன்படுத்தப்படுவதில்லை.

SMOD = 1 பாட் விகிதம் = 1/32 அலையியற்றி அதிர்வெண்

SMOD = 0 பாட விகிதம் = 1/64 அலையியற்றி அதிர்வெண்

முறை 3-இன் தொடர் வரிசைத் துறைமுகம் :

முறை 3-இன் பாட விகிதம் மாறக்கூடியது. (மாறி) இதன் பாட் விகிதம் முறை 1-ஐ ஒத்திருக்கும்.

தொடர் வரிசைத் தரவு மாறுதலுக்கான 8051-இன் திட்டநிரல் :

(Programming 8051 for serial data transfer) :

8051-க்குத் திட்டநிரலிடத், தரவினை தொடர் வரிசையில் செலுத்துவதற்குக் கீழ்க்கண்ட வரிசையில் செயல்களைச் செய்வதற்காக 8051-ஐ திட்டநிரலிட வேண்டும். அவை யாவன:

1) முறை 2-இல் நேரச்சுற்று 1-ஐ பயன்படுத்தி, பாட் விகிதத்தை அமைக்க TMOD பதிவேட்டை 20H என்ற மதிப்பைக் கொண்டு சுமையேற்ற வேண்டும்.

2) தொடர்வரிசைத் தரவு மாறுதலுக்காகத் தேவையான பாட விகிதத்தைப் பெற TH1-ஐ சுமையேற்ற வேண்டும்.

3) தொடர்வரிசை முறை 1-ஐ பயன்படுத்துவதற்காக SCON பதிவேட்டை 50H என்கிற மதிப்பைக் கொண்டு சுமையேற்ற வேண்டும்.

4) TR1 = 1 என அமைத்து, நேரச்சுற்று 1-ஐ தொடங்க வேண்டும்.

5) CLR T I கட்டளையைப் பயன்படுத்தி TI - ஐ காலி செய்யவேண்டும்.

6) எழுத்தினை எழுதி SBUF பதிவேட்டில் அனுப்ப வேண்டும்.

7) JNB TI, XXXX என்ற கட்டளையைப் பயன்படுத்தி, TI கொடி பிட்டைத் தடை செய்யவேண்டும்.

8) அடுத்த எழுத்தை மாற்றுவதற்காக வரிசை 5-க்குச் செல்லவேண்டும்.

தொடர் வரிசைத் தரவு பெறுவதற்கான 8051-இன் திட்டநிரல் : (Programming 8051 for Receiving serial data)

தரவினைத் தொடர் வரிசை முறையில் பெறுவதற்குக் கீழ்க்கண்ட வரிசையில் செயல்களைச் செய்வதற்காக, 8051-ஐ திட்டநிரல் வேண்டும்.

1) முறை 2-இல் நேரச்சுற்று 1-ஐ பயன்படுத்தி, பாட் விகிதத்தை அமைக்கப் பதிவேட்டை 20 என்ற மதிப்பைக் கொண்டு சுமையேற்ற வேண்டும்.

2) தொடர்வரிசைத் தரவு மாறுதலுக்காகத் தேவையான பாட விகிதத்தைப் பெற 1-ஐ சுமையேற்ற வேண்டும்.

3) தொடர்வரிசை முறை 1-ஐ பயன்படுத்துவதற்காகப் பதிவேட்டை 50 என்கிற மதிப்பைக் கொண்டு சுமையேற்ற வேண்டும்.

4) TR 1 = 1 என அமைத்து, நேரச்சுற்று 1-ஐ தொடங்க வேண்டும்.

5) CLR RI கட்டளையைப் பயன்படுத்தி RI - ஐ காலி செய்யவேண்டும்.

6) JNB RI, XXXX என்ற கட்டளையைப் பயன்படுத்தி RI கொடி பிட்டைத் தடை செய்யவேண்டும்.

7) RI = 1 எனில், SBUF பதிவேடு பைட்டை வைத்திருக்கும். இந்தப் பைட்டைச் சேமிக்கவேண்டும்.

8) அடுத்து எழுத்தைப் பெற வரிசை 5-க்குச் செல்லவேண்டும்.

5.18. முகவரியிடல் முறைகள் : (Addressing Modes)

முகவரியிடல் முறைகள் என்பது, முகவரிகள் கட்டளைகளில் எவ்வாறு குறிப்பிடப்படுகின்றன என்பதைக் குறிக்கும். பொதுவாகக் கட்டளைகளுக்குத் தேவையான தரவு, பதிவேடுகளில் அல்லது நினைவகத்தில் இருக்கும். இவற்றில் இருக்கின்ற தரவைப் பலவகையான முறைகளைப் பயன்படுத்தி அடையலாம். தரவை அடையும் வழிமுறைகள் முகவரியிடல் முறைகள் எனப்படும். 8051 நுண்கட்டுப்படுத்தி, ஐந்து வகையான முகவரியிடல் முறைகளைக் கொண்டுள்ளது. அவை யாவன:

1) பதிவேடு முகவரியிடல் **(Register Addressing)**
2) நேரடி பைட் முகவரியிடல் **(Direct byte Addressing)**
3) பதிவேடு மறைமுக முகவரியிடல் **(Register indirect Addressing)**
4) உடனடி முகவரியிடல் **(Immediate Addressing)**
5) பதிவேடு குறிப்பு முகவரியிடல் **(Register specific Addressing)**

பதிவேடு முகவரியிடல் :

இவ்வகையில், தேர்வு செய்யப்பட்டுள்ள வைப்பகத்தில் உள்ள R0 - R7 பதிவேடுகளைக் கொண்டு, இயமிகள் குறிப்பிடப்படுகின்றன. இதில் உள்ள கட்டளைகள் செயல்படுத்தப்படும்போது, தேர்ந்தெடுக்கப்பட்ட பதிவேடு வைப்பகத்தில் உள்ள எட்டுப் பதிவேடுகளில், ஒரே ஒரு பதிவேடு மட்டுமே ஒரு நேரத்தில் அடைய முடியும். திட்டநிரலாளர் பதிவேட்டில் உள்ள 4 மற்றும் 3 (வைப்பகத் தேர்வி) பிட்டுகளைப் பயன்படுத்தி, நான்கு வைப்பகங்களில் ஏதாவதொரு வைப்பகத்தை மட்டும் ஒரு நேரத்தில் தேர்ந்தெடுத்துக் கொள்ளலாம்.

ACC, B, DPTR மற்றும் CY (பூலியன் முறைப்படுத்திக்குரிய சேமக்கலம்) ஆகிய பதிவேடுகளாக முகவரியிடலாம். பதிவேடுகள் A, DPTR மற்றும் R0 - R7 இவைகள், செயற்குறிப்பின் ஒரு பகுதியாக இருக்கலாம். 8051-இல் உள்ள மற்ற பதிவேடுகள், நேரடி முகவரியிடல் முறையின்மூலம் முகவரியிடப்படலாம்.

செயற்குறிப்பு	செயல்
MOV A, Rr	பதிவேடு Rr- லுள்ள தரவைப் பதிவேடு A -க் கு நகர்த்தும்.
MOV Rr, A	பதிவேடு A - லுள்ள தரவைப் பதிவேடு Rr - க்கு நகர்த்தும்.
MOV A, # 0F1H	தரவு பைட் F1H ஐ உடனடியாகப் பதிவேடு A - க்குக் கொண்டு செல்லும்.

உதாரணம் : வைப்பகம் 2-லிருந்து பதிவேடு R3 மற்றும் R4 -இல் உள்ளவற்றைக் கூட்டு.

படி 1 பதிவேடு வைப்பகத்தைத் தேர்ந்தெடு.

MOV PSW, # 00001000B ; பதிவேடு வைப்பகம் 2-ஐ தேர்ந்தெடு.

படி 2 R3 மற்றும் R4 -இல் உள்ளவற்றைக் கூட்டு

MOV A, R3

ADD A, R4

நேரடி பைட் முகவரியிடல் :

இவ்வகையில் இயமி, 8 பிட் கொண்ட முகவரியின் மூலம் நேரடியாகக் குறிப்பிடப்படுகிறது. இதன்மூலம் உள்அகத் தரவு RAM (MSB-0) மற்றும் சிறப்புச் செயல் பதிவேடுகளை (MSB=1), நேரடியாக முகவரியிட முடியும்.

உதாரணம் :

MOV direct, A

இந்தக் கட்டளை, சேமக்கலத்திலுள்ள மதிப்பை நேரடியாக முகவரியிடப்பட்ட இருப்பிடத்திற்குக் கொண்டு செல்கின்றது.

குறிப்பு : கட்டளையிலுள்ள direct என்பது, உள்அகத் தரவு RAM அல்லது சிறப்புச் செயல்சார் பதிவேடுகளிலுள்ள இருப்பிடங்களை நேரடியாகக் குறிப்பிடுகின்ற 8-பிட் முகவரி ஆகும்.

உதாரணம் : 50H மற்றும் 51H இருப்பிடங்களில் உள்ளவற்றைக் கூட்டு.

MOV A. 50H : முகவரி 50H லிருந்து பைட்டைப் பதிவேடு Aக்குச் சுமையேற்று.

ADDA, 51H : 51H என்ற நினைவகஇருப்பிடத்தில் உள்ளவற்றை A-இல் உள்ளவற்றுடன் கூட்டு.

மறைமுக முகவரியிடல் முறை :

இந்த முறை முகவரியிடலில், இயமி இருக்கின்ற நினைவக இருப்பிடங்களின் முகவரி, பதிவேடுகளின்மூலம் கட்டளைகளில் குறிப்பிடப்படுகிறது. உள்அக மற்றும் புற RAM -களை முகவரியிடுவதற்கு, இம்முறையைப் பயன்படுத்திக் கொள்ளலாம்.

இயமி இருக்கின்ற நினைவக இருப்பிடங்களின் முகவரியைக் குறிப்பிடுவதற்கு, தேர்ந்தெடுக்கப்பட்ட பதிவேடு வைப்பகத்திலுள்ள R0 அல்லது R1 பதிவேடு அல்லது அடுக்குக் குறிமுள் பயன்படுத்தப்படுகிறது. இயமி இருக்கின்ற நினைவக இருப்பிடங்களின் 16 பிட் முகவரியைக் குறிப்பிடுவதற்குத் தரவுக் குறிமுள் (DPTR) பயன்படுத்தப்படுகிறது.

R0 அல்லது R1 பதிவேடுகளைக் கொண்டு உள்அக RAM - லுள்ள அனைத்து இருப்பிடங்களையும் (00 - 7FH), புறத் தரவு நினைவகத்திலுள்ள தாழ் 256 பைட இருப்பிடங்களையும் 8 பிட் கொண்ட முகவரியின் மூலம் முகவரியிடலாம்.

தரவு நினைவகத்திலுள்ள முழு 64 கி பைட் இருப்பிடங்களும், 16 பிட் DPTR பதிவேடின்மூலம் முகவரியிடப்படுகின்றது. சிறப்புச் செயல்சார் பதிவேடுகளை, மறைமுக முகவரியிடலைக் கொண்டு குறிப்பிட முடியாது.

உதாரணம் :

செயற்குறிப்பு	செயல்
MOV @ RP, #n	உடனடி பைட் n ஐ Rp -லுள்ள முகவரியில் வை.
MOV @ Rp, add	Address-இல் உள்ளவற்றை Rp -லுள்ள முகவரியில் வை.
MOV @ Rp, A	A-யிலுள்ள தரவை Rp -லுள்ள முகவரியில் வை.
MOV add, @ Rp	Rp-யிலுள்ள முகவரியில் உள்ளவற்றை Address-இல் வை.
MOV A, @ Rp	Rp-யினுள்ள முகவரியில் உள்ளவற்றை A -இல் வை.

உடன் முகவரியிடல் முறை :

இந்த முகவரியிடல் முறையில், இயமி ஒரு மாறிலி ஆகும். அதனால், கட்டளையில் மாறிலியை இணைக்க வேண்டும். # குறி என்பது, உடன் முகவரியிடல் முறை என்பதைக் குறிக்கும். இந்த முறையைப் பயன்படுத்தி உடன் தரவை நகர்த்தல், எண் கணக்கியல் மற்றும் வாதியல் செயல்களை உடன் தரவுடன் செய்தல் போன்ற செயல்களைச் செய்யலாம்.

உதாரணம் :

MOV A, # 52

இந்தக் கட்டளை, 52 என்ற தரவைச் சேமக்கலத்தில் சுமையேற்றம் செய்யும்.

செயற்குறிப்பு	செயல்
MOV Rr, # n	8 பிட் எண். n – ஐ பதிவேடு Rr –இல் வை
MOV A, # n	8 பிட் எண். n – ஐ பதிவேடு A –இல் வை
MOV DPTR, # nn	16 பிட் எண். nn – ஐ பதிவேடு DPTR –இல் வை

பதிவேடு குறிப்பு முகவரியிடல் :

இது கட்டளையின் உள்ளேயே உள்ளது. இது சேமக்கலம் அல்லது DPTR போன்ற குறிப்பிட்ட பதிவேடைக் குறிப்பிடுகிறது.

உதாரணம் :

SWAP A:

அடைவமைத்த முகவரியிடல் :

அடைவமைத்த முகவரியிடலில், திட்டநிரல் நினைவகத்தை மட்டுமே அடைய முடியும். இந்த முகவரியிடலைப் பயன்படுத்தித் திட்டநிரல் நினைவகத்தை மட்டும் அடைய முடியும். அதில் உள்ள இருப்பிடங்களில் இருந்து, தரவைப் படிக்க மட்டும் செய்துகொள்ளலாம். பொதுவாக, அடைவமைத்த முகவரியிடல், திட்டநிரல் நினைவகத்தில் உள்ள நோக்கீட்டு அணிப்பட்டியைப் படிப்பதற்குப் பயன்படுகிறது.

உதாரணம் : MOVC A, @ A + DPTR

இந்தக் கட்டளை, DPTR மற்றும் சேமக்கலம் ஆகிய இரண்டில் உள்ள மதிப்புகளைக் கூட்டுகிறது. அந்தக் கூட்டுத்தொகையை முகவரியாகப் பயன்படுத்துகிறது. அதிலிருந்து பைட்டானது சேமக்கலத்திற்கு நகருகிறது.

5.19. 8051 கட்டளைத் தொகுப்பு : (Instruction Set)

i) **தரவு மாற்றும் கட்டளைகள் : (Data Transfer Instructions)**

தரவு மற்றும் கட்டளைகள், மூன்று வகைகளாகப் பிரிக்கப்படுகின்றன.

1. பொது நோக்குக் கட்டளைகள்.
2. சேமக்கலம் குறித்த கட்டளைகள்.
3. முகவரி – நோக்கக் கட்டளைகள்

1. பொது நோக்குத் தரவு மாற்றக் கட்டளைகள் :

இரண்டாவது இயமியால் குறிக்கப்பட்ட பைட் மாறி, முதல் இயமியால் குறிப்பிடப்பட்ட இடத்தில் வைக்கப்படும். மூல பைட் மாறாது, அப்படியே இருக்கும். வேறு அல்லது பிற பதிவேடு அல்லது கொடியில் மாற்றம் இருக்காது.

i) MOV கட்டளைகள்

ii) PUSH கட்டளை

iii) POP கட்டளை

ஆகிய கட்டளைகள் இவ்வகையைச் சாரும்.

i) MOV A, Rn (பதிவேடிலிருந்து சேமக்கலத்திற்கு நகர்த்து)

செயல்: (A) ⟵ (Rn)

பதிவேடு Rn–இல் உள்ள மதிப்பின் நகலைப் பதிவேடு A–இல் வை.

MOV A, R0

ii) MOV A, direct (நேரடி பைட்டிலிருந்து சேமக்கலத்திற்கு நகர்த்து)

MOV A, 30 H

செயல்: (A) ⟵ (direct) நேரடியாக முகவரியிடப்பட்ட இருப்பிடத்தில் இருக்கின்ற மதிப்பின் நகலைச் சேமக்கலத்திற்கு நகர்த்து.

iii) MOV A, @ Ri (மறைமுக RAM –லிருந்து சேமக்கலத்திற்கு நகர்த்து).

(A) ⟵ (Ri)

உதாரணம் : MOV A, @ R1 பதிவேடு Ri குறிப்பிடுகின்ற நினைவக இருப்பிடங்களில் உள்ள மதிப்பின் நகலைச் சேமக்கலத்திற்கு நகர்த்து.

iv) MOV A, # data (உடனடித் தரவைச் சேமக்கலத்திற்கு நகர்த்து)

A ⟵ # data

உதாரணம் : MOV A, # 30H

கட்டளையுடன் கொடுக்கப்பட்ட தரவினைச் சேமக்கலத்தில் வை.

v) MOV Rn, A (சேமக்கலத்திலிருந்து பதிவேட்டிற்கு நகர்த்து)

உதாரணம் : (Rn) ⟵ (A)

MOV R2, A

தேர்ந்தெடுக்கப்பட்ட பதிவேடு/வைப்பகத்திலுள்ள R2 பதிவேட்டில், சேமக்கலத்தில் உள்ளவற்றை வை என்பதைக் கட்டளை குறிக்கிறது.

vi) MOV Rn, direct (நேரடிப் பைட்டைச் சேமக்கலத்திற்கு நகர்த்து)

(Rn) ⟵ (direct)

உதாரணம் : MOV R1, 40H

நினைவக முகவரி 40H–இல் உள்ளவற்றைத் தேர்ந்தெடுக்கப்பட்ட பதிவேடு வைப்பகத்திலுள்ள பதிவேட்டில் வை என்பதை, இந்தக் கட்டளை குறிக்கும்.

vii) MOV Rn, # data (நேரடித் தரவை பதிவேட்டிற்கு நகர்த்து)

(Rn) ⟵ # data

உதாரணம்: MOV R2, # 20H. இந்தக் கட்டளை தேர்ந்தெடுக்கப்பட்ட பதிவேடு வைப்பகத்தில் உள்ள பதிவேடு R2 -இல் 20H ஐ சுமையேற்றும்.

viii) MOV direct, A (சேமக்கலத்திலிருந்து நேரடி பைட்டிற்கு நகர்த்து).

(direct) ⟵ (A)

உதாரணம் : MOV 20H, A: இந்தக் கட்டளையானது, சேமக்கலத்தில் உள்ளவற்றை, இந்தக் கட்டளையில் (20H) குறிக்கப்பட்ட நேரடி நினைவக முகவரியில் வைக்கும்.

ix) MOV direct, Rn (பதிவேடிலிருந்து நேரடி பைட்டிற்கு நகர்த்து),

(direct) ⟵ (Rn).

உதாரணம்: MOV 30H, R2 : இந்தக் கட்டளை தேர்ந்தெடுக்கப்பட்ட பதிவேடு

வைப்பகத்திலுள்ள பதிவேடு R2-இல் உள்ளவற்றை, இந்தக் கட்டளையில் (30H) குறிக்கப்பட்டுள்ள நேரடி நினைவக முகவரியில் வைக்கும்.

x) MOV (direct), (direct) (நேரடி பைட்டிலிருந்து நேரடி பைட்டிற்கு நகர்த்து)

(direct) ⟵ (direct)

உதாரணம்: MOV 20H, 40H: இந்தக் கட்டளை, நினைவக இருப்பிடத்தில் (40H) உள்ளவற்றை நினைவக இருப்பிடம் (20H) -இல் வைக்கும்.

xi) MOV direct, @ Ri

(direct) ⟵ ((Ri))

உதாரணம் : MOV 20H, @ R3

இந்தக் கட்டளை, தேர்ந்தெடுக்கப்பட்ட பதிவேடு வைப்பகத்தினுடைய பதிவேடு R3 -இல் கொடுக்கப்பட்ட நினைவக இருப்பிடத்தின் முகவரியில் உள்ளவற்றை இருப்பிடம் 20H-இல் உள்ள நினைவக இருப்பிடத்தின் முகவரியில் வைக்கும்.

xii) MOV direct, # data

(direct) ⟵ # data

உதாரணம் : MOV 30H, # 12H

இந்தக் கட்டளையானது, கட்டளையில் (12H) கொடுக்கப்பட்ட தரவை நினைவக இருப்பிடத்தின் முகவரி 30H-இல் வைக்கும்.

xiii) MOV @ Ri, A

((Ri)) ⟵(A)

உதாரணம் : MOV @ R1, A

இந்தக் கட்டளை, சேமக்கலத்தில் உள்ளவற்றைத் தேர்ந்தெடுக்கப்பட்ட பதிவேடு வைப்பகத்தின் R1–இல் கொடுக்கப்பட்டுள்ள நினைவக இருப்பிடத்தின் முகவரியில் வைக்கும்.

MOV @ Ri, direct

((Ri)) ⟵ (direct)

உதாரணம் : MOV @ R2, 30H இந்தக் கட்டளை, நினைவக இருப்பிடத்தில் உள்ளவற்றைத் தேர்ந்தெடுக்கப்பட்ட பதிவேடு வைப்பகத்தின் R2 பதிவேடால் குறிக்கப்பட்ட நினைவக இருப்பிடத்தின் முகவரியில் வைக்கும்.

MOV @ Ri, # data

((Ri)) ⟵ # data

உதாரணம் : MOV @ R2, # 30H: இந்தக் கட்டளை, தேர்ந்தெடுக்கப்பட்ட பதிவேடு வைப்பகத்தின் R2 பதிவேடால் குறிக்கப்பட்ட நினைவக இருப்பிடத்தின் முகவரியில் 30H–ஐ சுமையேற்றும்.

MOV A, ACC என்பது செல்லக்கூடிய கட்டளை அல்ல.

MOV DPTR, # data 16 பைட்டுகள் : 3 சுழற்சிகள் : 2

செயல் : (தரவுக் குறிமுள்ளை 16 பிட் மாறிலி கொண்டு சுமையேற்று)

விளக்கம் : தரவுக் குறிமுள், குறிப்பிடப்பட்ட 16 பிட் மாறிலி கொண்டு சுமையேற்று. கட்டளையின் இரண்டாவது மற்றும் மூன்றாவது பைட்டுகளில், 16 பிட் மாறிலி சுமையேற்றப்படுகிறது. இரண்டாவது பைட் உயர் – வரிசை (DPH) பைட்டையும், மற்றும் மூன்றாவது பைட் தாழ் வரிசை பைட்டையும் (DPL) வைத்திருக்கும். கொடிகள் எவற்றிலும் மாற்றம் இருக்காது. இந்தக் கட்டளை மட்டுமே தரவின் 16 பிட்டை ஒரே நேரத்தில் நகர்த்துகிறது.

உதாரணம் : MOV DPTR, # 1234H

இந்தக் கட்டளை, 1234 மதிப்பைத் தரவுக் குறிமுள்ளில் சுமையேற்றும். DPH - 12H–யும் DPL - 34H –யும் வைத்திருக்கும்.

செயல்: MOV

(DPTR) ⟵ # data 15-0

DPH ⟵ # data 15-8 DPL ⟵ # data 7-0

MOV C A, @ A+ < base - reg> பைட்டுகள் : 1 சுழற்சிகள் : 2

செயல் : குறியீடு பைட்டுகளை நகர்த்து.

MOVC A, @ A+DPTR

இந்தக் கட்டளை, சேமக்கலத்தைத் திட்டநிரல் நினைவகத்தில் உள்ளவற்றைக் கொண்டு சுமையேற்றும். திட்டநிரல் நினைவகத்தின் முகவரி, சேமக்கலம் மற்றும் DPTR பதிவேடு ஆகியவற்றில் உள்ளவற்றின் கூட்டுத்தொகையைக் கொடுக்கும்.

செயல் : (A) ⟵ (A) + (DPTR)

MOVC A, @ A+PC

இந்தக் கட்டளை, சேமக்கலத்தைத் திட்டநிரல் நினைவகத்தில் உள்ளவற்றைக் கொண்டு சுமையேற்றும். திட்டநிரல் நினைவகத்தின் முகவரி, சேமக்கலம் மற்றும் திட்டநிரல் எண்ணி ஆகியவற்றில் உள்ளவற்றின் கூட்டுத்தொகையைக் கொடுக்கும். திட்டநிரல் எண்ணியில் தற்சமயம் உள்ளவை, கூட்டுவதற்கு முன்பாக 1ஆல் அதிகரிக்கப்படும்.

செயல் : (PC) ← (PC) + 1

(A) ← ((A) + (PC))

MOVX <dest - byte>, <src - byte> பைட்டுகள் : 1 சுழற்சிகள் : 2

செயல் : புறவழியாக நகர்த்து.

MOVX A, @ Ri

(A) ← ((Ri))

இந்தக் கட்டளை, புற நினைவகத்தில் உள்ளவற்றைச் சேமக்கலத்தில் வைக்கும். புற நினைவகத்தின் முகவரி, பதிவேடால் கொடுக்கப்படும்.

MOVX A, @ DPTR :

இந்தக் கட்டளை, புற நினைவகத்தில் உள்ளவற்றைச் சேமக்கலத்தில் வைக்கும். புற நினைவகத்தின் முகவரி, பதிவேடால் கொடுக்கப்படும்.

செயல் : (A) ← ((DPTR))

MOVX @ Ri, A : இந்தக் கட்டளை, சேமக்கலத்தில் உள்ளவற்றைப் புற நினைவகத்தில் வைக்கும். புற நினைவகத்தின் முகவரி, பதிவேடால் கொடுக்கப்படும்.

செயல் : ((Ri)) ← A

MOVX @ DPTR, A : இந்தக் கட்டளை, சேமக்கலத்தில் உள்ளவற்றைப் புற நினைவகத்தில் வைக்கும். புற நினைவகத்தின் முகவரி, DPTR பதிவேடால் கொடுக்கப்படும்.

செயல் : ((DPTR)) ← A

PUSH direct

செயல்: அடுக்கிற்குத் தள்ளு பைட்டுகள் : 2 சுழற்சிகள் : 2

உதாரணம் : குறுக்கீடு நிரற்செயன்முறையில் நுழையும்போது, அடுக்குக் குறிமுள் 09H–ஐ வைத்திருக்கும். தரவுக் குறிமுள் 0123H என்கிற மதிப்பை வைத்திருக்கும். கட்டளை வரிசை,

PUSH DPL

PUSH DPH

ஆகியவை, அடுக்குக் குறிமுள்ளை 0BH என்று அமைக்கும். 23H மற்றும் 01H ஆகியவற்றை, உள்அக RAM இருப்பிடங்கள், 0AH மற்றும் 0BH ஆகியவற்றில் சேமிக்கின்றன.

செயல் : (SP) ← (SP) + 1

((SP)) ← (direct)

POP direct

செயல்: அடுக்கிலிருந்து எடு பைட்டுகள் : 2 சுழற்சிகள் : 2

விளக்கம் : அடுக்குக் குறிமுள்ளால் முகவரியிடப்பட்ட உள்அக RAM, இருப்பிடத்தில் உள்ளவற்றைப் படித்துப் பின், அடுக்குக் குறிமுள் மதிப்பு ஒன்று குறைக்கப்படும். பழக்கப்பட்ட மதிப்பு, பின்னர் நேரடியாகக் குறிப்பிடப்பட்ட முகவரி பைட்டிற்கு மாறும். கொடிகளில் எந்த மாற்றமும் இருக்காது.

செயல் : (direct) ← ((SP))
(SP) ← (SP) - 1

XCHA, <byte> பைட்டுகள் : 1/2 சுழற்சிகள் : 1

செயல் : சேமக்கலத்தைப் பைட் மாறியுடன் பரிமாற்றும்.

உதாரணம் : R0 பதிவேடு 20H என்கிற முகவரியை வைத்திருக்கும். சேமக்கலம் 3FH (00111111B) என்ற மதிப்பை வைத்திருக்கும். உள்அக RAM இருப்பிடம் 20H ஆனது 75H (01110101B) என்ற மதிப்பை வைத்திருக்கும்.

XCH A, @ R0 என்கிற கட்டளையில் RAM இருப்பிடம் 20H ஆனது. 3FH (00111111B) மற்றும் 75H (01110101B) ஆகிய மதிப்புகளைச் சேமக்கலத்தில் வைத்திருக்கும்.

XCH A, Rn

செயல் : (A) ⇆ (Rn)

உதாரணம் : XCH A, R0 இந்தக் கட்டளை, சேமக்கலத்தில் உள்ளவற்றைத் தேர்ந்தெடுத்த பதிவேடு வைப்பகத்தின் R0 பதிவேடில் உள்ளவற்றுடன் பரிமாற்றம் செய்கிறது.

XCH A, direct

செயல் : (A) ⇆ (direct)

உதாரணம் : XCH A, 20H இக்கட்டளை, சேமக்கலத்தில் உள்ளவற்றை நினைவகத்தில் உள்ளவற்றுடன் பரிமாற்றம் செய்கிறது. நினைவகத்தின் முகவரி, கட்டளையிலே (20H) கொடுக்கப்பட்டுள்ளது.

X CH A, @ Ri

செயல் : (A) ⇆ (Ri)

உதாரணம் : MOV A, # R2: இந்தக் கட்டளை, சேமக்கலத்தில் உள்ளவற்றை நினைவக இருப்பிடத்தில் உள்ளவற்றுடன் உடன் பரிமாற்றம் செய்கிறது. நினைவக இருப்பிடத்தின் முகவரி, தேர்ந்தெடுத்த பதிவேடு வைப்பகத்தின் R2 பதிவேட்டில் உள்ளவற்றைக் கொடுக்கும்.

X CHD A, @ Ri செயல் : இலக்கப் பரிமாற்றம் பைட்டுகள் : 1 சுழற்சிகள் : 1

உதாரணம் : R0 பதிவேடு முகவரி 20Hஐ வைத்திருக்கும். சேமக்கலம் 36H

(00110110B) என்கிற மதிப்பை வைத்திருக்கும். உள்அக RAM இருப்பிடம் 20H ஆனது 20H (00100000B) மற்றும் 75H (01110101B) ஆகிய மதிப்புகளை வைத்திருக்கும்.

X CHD A, @ R0

இந்தக் கட்டளை, RAM இருப்பிடம் 20H ஆனது 76H (01110110B) மற்றும் 35H (00110101B) ஆகிய மதிப்புகளைச் சேமக்கலத்தினுள் வைத்திருக்கும்.

செயல் : $(A_{3-0}) \rightleftarrows (R_{i3-0})$

எண்கணக்கியல் கட்டளைகள் : (Arithmetic Instructions)

ADD A, <Src - byte>

செயல் : கூட்டுதல் பைட்டுகள் : 1/2 சுழற்சிகள் : 1

உதாரணம் : சேமக்கலம் 89H (10001001B) மற்றும் பதிவேடு 0 ஆனது 95H (10010101B) வைத்திருக்கும்.

ADD A, R0

இந்தக் கட்டளை, 1E (00011101B) – ஐ சேமக்கலத்தில் விடும். AC கொடியைக் காலியாக்கி விடும் மற்றும் இரண்டு சுமைக் கொடிகளையும் அமைவு நிலை (1)-யில் வைக்கும்.

ADD A, Rn

செயல் : (A) ⟵ (A) + (Rn)

உதாரணம் : ADD A, R2

R2 பதிவேடில் உள்ள மதிப்பானது, A பதிவேடில் உள்ள மதிப்புடன் கூட்டப்படும். இதன் முடிவு A பதிவேடில் சேமிக்கப்படுகிறது.

ADD A, direct

செயல் : (A) ⟵ (A) + (direct)

உதாரணம் : ADD A, 20H கட்டளையில் நேரடியாகக் குறிப்பிடப்பட்டுள்ள நினைவக இருப்பிடத்தில் உள்ள மதிப்பு A பதிவேடில் உள்ள மதிப்புடன் கூட்டப்படும். இதன் முடிவு, A பதிவேடில் சேமிக்கப்படுகிறது.

ADD A, @ Ri

செயல் : (A) ⟵ (A) + ((Ri))

Ri பதிவேடில் குறிப்பிடுகின்ற உள்அக RAM முகவரி இருப்பிடங்களில் உள்ள மதிப்பானது, A பதிவேடில் உள்ள மதிப்புடன் கூட்டப்படும். இதன் முடிவு, A பதிவேடில் சேமிக்கப்படுகின்றது.

உதாரணம் : ADD, A @ R2

ADD A, # data

செயல் : (A) ⟵ (A) + # data

கட்டளையில் இரண்டாவது பைட்டில் குறிப்பிடப்பட்டுள்ள மதிப்பானது, பதிவேடு A-யில் உள்ள மதிப்புடன் கூட்டப்படும். இதன் முடிவு, A பதிவேடில் சேமிக்கப்படுகின்றது.

உதாரணம் : ADD A, # 20H

ADDC A, <Src - byte >

செயல் : சுமையுடன் கூட்டு. பைட்டுகள் : 1/2 சுழற்சிகள் : 1

ADDC A, Rn

செயல் : (A) ⟵ (A) + (C) + (Rn)

Rn பதிவேடில் உள்ள மதிப்பும் மற்றும் சுமைக் கொடியில் உள்ள மதிப்பும் A பதிவேடில் உள்ள மதிப்புடன் கூட்டப்படுகின்றது. இதன் முடிவு, பதிவேடு A-இல் சேமிக்கப்படுகின்றது.

ADDC A, direct

செயல் : (A) ⟵ (A) + (C) + (direct)

கட்டளையில் நேரடியாகக் குறிப்பிடப்பட்டுள்ள முகவரியில் உள்ள மதிப்பும் மற்றும் சுமைக் கொடியில் உள்ள மதிப்பும், A பதிவேடில் உள்ள மதிப்புடன் கூட்டப்படும். இதன் முடிவு, A பதிவேடில் சேமிக்கப்படுகின்றது.

ADDC A, @ Ri

செயல் : (A) ⟵ (A) + (C) + ((Ri))

Ri பதிவேடு குறிப்பிடப்படுகின்ற உள்அக RAM-யின் நினைவக முகவரியில் உள்ள மதிப்பும் மற்றும் சுமைக் கொடியில் உள்ள மதிப்பும், சேமக்கலத்தில் உள்ள மதிப்புடன் கூட்டப்படும். இதன் முடிவு, A பதிவேடில் சேமிக்கப்படுகின்றது.

ADDC A, # data

(A) ⟵ (A) + (C) + # data

கட்டளையின் இரண்டாவது பைட்டில் குறிப்பிடப்பட்டுள்ள மதிப்பும் மற்றும் சுமைக் கொடியில் உள்ள மதிப்பும், A பதிவேடில் உள்ள மதிப்புடன் கூட்டப்படுகின்றது. இதன் முடிவு, A பதிவேடில் சேமிக்கப்படுகிறது.

SUBB A, <Src - byte>

செயல் : இரவல் உடன் கழித்தல் பைட்டுகள் : 2 சுழற்சிகள் : 1

SUBB A, Rn

செயல் : (A) ⟵ (A) - (C) - (Rn)

Rn பதிவேடில் உள்ள மதிப்பும் மற்றும் சுமைக் கொடியில் உள்ள மதிப்பும், பதிவேடு A-இல் உள்ள மதிப்பிலிருந்து கழிக்கப்படுகிறது. இதன் முடிவு, A பதிவேடில் சேமிக்கப்படுகிறது.

SUBB A, direct

செயல் : (A) ⟵ (A) - (C) - (direct)

கட்டளையில் நேரடியாகக் குறிப்பிடப்பட்டுள்ள முகவரியில் உள்ள மதிப்பும் மற்றும் சுமைக் கொடியில் உள்ள மதிப்பும், பதிவேடு A–இல் உள்ள மதிப்பில் இருந்து கழிக்கப்படுகிறது. இதன் முடிவு, பதிவேடு A–இல் சேமிக்கப்படுகிறது.

SUBB A, @ Ri

செயல் : (A) ⟵ (A) - (C) - ((Ri))

பதிவேடு Ri குறிப்பிடுகின்ற உள்அக RAM –இன் நினைவக இருப்பிடத்தில் உள்ள மதிப்பும் மற்றும் சுமைக் கொடியில் உள்ள மதிப்பும் பதிவேடு A–இல் உள்ள மதிப்பிலிருந்து கழிக்கப்படுகிறது. இதன் முடிவு பதிவேடு A–இல் சேமிக்கப்படுகிறது.

SUBB A, # data

செயல் : (A) ⟵ (A) - (C) - # data

கட்டளையின் இரண்டாவது பைட்டில் குறிப்பிடப்பட்டுள்ள மதிப்பும் மற்றும் சுமைக் கொடியில் உள்ள மதிப்பும், பதிவேடு A–இல் உள்ள மதிப்பிலிருந்து கழிக்கப்படுகிறது. இதன் முடிவு பதிவேடு A–இல் சேமிக்கப்படுகிறது.

INC <byte>

செயல் : உயர்த்தப்படுதல் பைட்டுகள் : 1/2 சுழற்சிகள் : 1

INC A

செயல் : (A) ⟵ (A) + 1

INC Rn

செயல் : (Rn) ⟵ (Rn) + 1

உதாரணம் : INC R2

பதிவேடு R2–இல் உள்ள மதிப்பானது ஒன்று அதிகரிக்கின்றது.

INC direct

செயல் : (direct) ⟵ (direct) + 1

உதாரணம் : INC 20H

கட்டளையில் குறிப்பிடப்பட்டுள்ள நேரடி பைட் இருப்பிடத்தில் உள்ள மதிப்பானது, ஒன்று அதிகரிக்கின்றது.

INC @ Ri

செயல் : ((Ri)) ⟵ ((Ri)) + 1

பதிவேடு Ri குறிப்பிடுகின்ற உள்அக RAM முகவரியில் உள்ள மதிப்பானது, ஒன்று அதிகரிக்கின்றது.

உதாரணம் : INC @ R2

INC DPTR

செயல் : தரவுக் குறிமுள்ளை அதிகரித்தல்

பைட்டுகள் : 1 சுழற்சிகள் : 2

விளக்கம் : 16–பிட் தரவு குறிமுள் ஒன்று அதிகரிக்கப்படுகிறது.

செயல் : (DPTR) ⟵ (DPTR) + 1

DEC byte செயல் : குறைக்கப்படுதல்
பைட்டுகள் : 1/2 சுழற்சிகள் : 1
DEC A
செயல் : (A) ⟵ (A) - 1
DEC Rn
செயல் : (Rn) ⟵ (Rn) - 1

பதிவேடு Rn-இல் உள்ள மதிப்பானது, ஒன்று குறைக்கின்றது.
DEC direct
செயல் : (direct) ⟵ (direct) - 1

கட்டளையில் குறிப்பிடப்பட்டுள்ள நேரடி பைட் முகவரியில் உள்ள மதிப்பானது, ஒன்று குறைக்கின்றது.
DEC @ Ri
செயல் : ((Ri)) ⟵ ((Ri)) - 1

பதிவேடு Ri குறிப்பிடுகின்ற உள்அக RAM முகவரியில் உள்ள மதிப்பானது ஒன்று குறைகின்றது.

MUL AB செயல் : பெருக்குப் பைட்டுகள் : 1 சுழற்சிகள் : 4

சேமக்கலத்தில் உள்ள மதிப்பு, B பதிவேடில் உள்ள மதிப்புடன் பெருக்கப்படுகின்றது. கிடைக்கப்பெறுகின்ற முடிவின் தாழ் பைட், சேமக்கலத்திலும் மற்றும் உயர்ந்த பைட் பதிவேடு B-யிலும் சேமிக்கப்படுகின்றன.

முடிவின் உயர் பைட் சுழியாக இருந்தால், OV கொடி சுழியாகி விடும் (0). மாறாகச் சுழி இல்லையெனில், கொடி அமைவு நிலையை (1) அடையும். பெருக்கல் செயலினால் சுமைக் கொடி சுழியாகி விடும் மற்றும் துணைச் சுமைக் கொடி பாதிக்கப்படுவதில்லை.

DIV AB செயல் : வகுத்தல் பைட்டுகள் : 1 சுழற்சிகள் : 4
DIV AB

சேமக்கலத்தில் உள்ள மதிப்பானது, பதிவேடு B-இல் உள்ள மதிப்பினால் வகுக்கப்படுகிறது. கிடைக்கும் முடிவின் ஈவு சேமக்கலத்திலும் மீதி பதிவேடு B-யிலும் சேமிக்கப்படுகின்றன.

ஏதாவதொரு மதிப்பினைச் சுழி (0) கொண்டு வகுத்தால்; முடிவு சரியாக இருக்காது. இதன்மூலம் தேவையற்ற முடிவுகள், A மற்றும் B பதிவேடுகளில் கிடைக்கப்பெறும், மற்றும் OV கொடி அமைவு நிலைக்குச் (1) செல்லும். சரியான முறையில் வகுத்தல் நடைபெற்றால், OV கொடி காலியாகி (0) விடும். வகுத்தல் செயலினால் CY கொடி காலியாகி (0) விடும் மற்றும் AC கொடி பாதிக்கப்படுவதில்லை.

DA A

செயல் : கூட்டல் பைட்டுகளுக்கான தசம எண்ணாக மாற்றும் சேமக்கலம்.

இந்தக் கட்டளை, சேமக்கலத்தில் உள்ள 8 - பிட் கொண்ட மதிப்பை 4 பிட்டுகள் கொண்ட இரண்டு BCD எண்களாக மாற்றுகிறது. சேமக்கலத்தின் 3–0களின் பிட் மதிப்பு, 9–ஐ விட (XXXX1010 - XXXX 1111) அதிகமாக இருந்தாலோ அல்லது AC கொடி அமைவு நிலையில் இருந்தாலோ, 6 ஆனது பதிவேடு A- யில் உள்ள மதிப்புடன் கூட்டப்படுகின்றது.

சேமக்கலத்தில் உள்ள 7–4 பிட்டுகளின் மதிப்பு 9–ஐ விட அதிகமாக இருந்தாலோ அல்லது சுமைக்கொடி அமைவுநிலையில் இருந்தாலோ, 6 சேமக்கலத்தில் உள்ள உயர் வரிசை நிப்பில் மதிப்புடன் கூட்டப்படுகிறது.

குறிப்பு : கட்டளை DAA, ஒரு கூட்டல் செயல் முடிந்து சேமக்கலத்தில் கிடைக்கப்பெறுகின்ற எண்ணினை மட்டுமே, அதற்குச் சமமான BCD எண்ணாக மாற்றுகின்ற தன்மை கொண்டது. சேமக்கலத்தில் இருக்கின்ற சாதாரண எண்ணினையோ அல்லது கழித்தல் செயல் முடிந்து கிடைக்கப்பெறுகின்ற எண்ணினையோ, BCD ஆக கண்டிப்பாக மாற்றாது.

பிட் மட்டத்தில் எண் கணக்கியல் கட்டளைகள் எதுவும் இல்லை.

வாதியல் கட்டளைகள்:

ANL <dest - byte> , <Src - byte>

செயல் : வாதியல் AND பைட்டுகள் : 1/2/3 சுழற்சிகள் : 1

ANL A, Rn

செயல் : (A) ← (A) ^ (Rn)

பதிவேடு Rn–இல் உள்ள மதிப்பு, சேமக்கலத்தில் உள்ள மதிப்புடன் வாதியல் முறையில், AND செய்யப்படுகிறது. இதன் முடிவு, சேமக்கலத்தில் சேமிக்கப்படுகிறது.

ANL A, direct

செயல் : (A) ← (A) ^ (direct)

கட்டளையில் நேரடியாகக் குறிப்பிடப்பட்டுள்ள முகவரியில் உள்ள மதிப்பானது, சேமக்கலத்தில் உள்ள மதிப்புடன் வாதியல் முறையில் AND ஆகின்றது. இதன் முடிவு, சேமக்கலத்தில் சேமிக்கப்படுகிறது.

ANL A, @ Ri

செயல் : (A) ⟵ (A) ^ ((Ri))

பதிவேடு Ri குறிப்பிடுகின்ற உள்அக RAM-இன் நினைவக முகவரியில் உள்ள மதிப்பானது, சேமக்கலத்தில் உள்ள மதிப்புடன் வாதியல் முறையில் AND ஆகின்றது. இதன் முடிவு, சேமக்கலத்தில் சேமிக்கப்படுகிறது.

ANL A, # data

செயல்: (A) ⟵ (A) ^ # data

கட்டளையின் இரண்டாவது பைட்டில் குறிப்பிடப்பட்டுள்ள மதிப்பானது, சேமக்கலத்தில் உள்ள மதிப்புடன் வாதியல் முறையில் AND ஆகின்றது. இதன் முடிவு, சேமக்கலத்தில் சேமிக்கப்படுகிறது.

ANL direct, A

செயல் : (direct) ⟵ (direct) ^ (A)

சேமக்கலத்தில் உள்ள மதிப்பானது, கட்டளையில் நேரடியாகக் குறிப்பிடப்பட்டுள்ள நினைவக இருப்பிடத்தில் உள்ள மதிப்புடன் வாதியல் முறையில் AND ஆகின்றது. இதன் முடிவு, அதே இருப்பிடத்தில் சேமிக்கப்படுகின்றது.

ANL direct, # data

செயல் : (direct) ⟵ (direct) ^ # data

கட்டளையின் இரண்டாவது பைட்டில் குறிப்பிடப்பட்டுள்ள மதிப்பானது, கட்டளையில் நேரடியாகக் குறிப்பிடப்பட்டுள்ள நினைவக இருப்பிடத்தில் உள்ள மதிப்புடன் வாதியல் முறையில் AND ஆகின்றது. இதன் முடிவு, அதே நினைவக இருப்பிடத்தில் சேமிக்கப்படுகிறது.

ORL <dest - byte> < Src - byte>

செயல் : பைட் மாறிகளுக்கான வாதியல் - OR

பைட்டுகள் : 1/2/3 சுழற்சிகள் : 1/2

ORL A, Rn

செயல் : (A) ⟵ (A) V (Rn)

பதிவேடு Rn-யில் உள்ள மதிப்பு, சேமக்கலத்தில் உள்ள மதிப்புடன் வாதியல் முறையில் OR செய்யப்படுகிறது. இதன் முடிவு, சேமக்கலத்தில் சேமிக்கப்படுகிறது.

ORL A, direct

செயல் : (A) ⟵ (A) V (direct)

கட்டளையில் நேரடியாகக் குறிப்பிடப்பட்டுள்ள நினைவக இருப்பிடத்தில் உள்ள மதிப்பு, சேமக்கலத்தில் உள்ள மதிப்புடன் வாதியல் முறையில் OR செய்யப்படுகிறது. இதன் முடிவு, சேமக்கலத்தில் சேமிக்கப்படுகிறது.

ORL A, @ Ri

செயல் : (A) ⟵ (A) V ((Ri))

பதிவேடு Ri குறிப்பிடுகின்ற உள்அக RAM -இன் நினைவக இருப்பிடத்தில் உள்ள மதிப்பு, சேமக்கலத்தில் உள்ள மதிப்புடன் வாதியல் முறையில் OR ஆகின்றது. இதன் முடிவு, சேமக்கலத்தில் சேமிக்கப்படுகிறது.

ORL A, # data

செயல் : (A) ← (A) v # data

கட்டளையின் இரண்டாவது பைட்டில் குறிப்பிடப்பட்டுள்ள மதிப்பு, சேமக்கலத்தில் உள்ள மதிப்புடன் வாதியல் முறையில் OR ஆகின்றது. இதன் முடிவு, சேமக்கலத்தில் சேமிக்கப்படுகிறது.

ORL direct, A

செயல் : (direct) ← (direct) V (A)

சேமக்கலத்தில் உள்ள மதிப்பு, கட்டளையில் நேரடியாகக் குறிப்பிடப்பட்டுள்ள நினைவக இருப்பிடத்தில் உள்ள மதிப்புடன் வாதியல் OR ஆகின்றது. இதன் முடிவு அதே நினைவக இருப்பிடத்தில் சேமிக்கப்படுகிறது.

ORL direct, # data

(direct) ← (direct) V # data

கட்டளையின் இரண்டாவது பைட்டில் குறிப்பிடப்பட்டுள்ள மதிப்பு கட்டளையில் நேரடியாகக் குறிப்பிடப்பட்டுள்ள நினைவக இருப்பிடத்தில் உள்ள மதிப்புடன் வாதியல் முறையில் OR ஆகின்றது. இதன் முடிவு, அதே இருப்பிடத்தில் சேமிக்கப்படுகிறது.

XRL <dest - byte> , <Src - byte>

செயல் : பைட் மாறிக்கான வாதியல் X OR

பைட்டுகள் : 1/2 சுழற்சிகள் : 1

XRL A, Rn

செயல் : (A) ← (A) (Rn)

பதிவேடு Rn -இல் உள்ள மதிப்பு, சேமக்கலத்தில் உள்ள மதிப்புடன் வாதியல் முறையில் XOR ஆகின்றது. இதன் முடிவு, சேமக்கலத்தில் சேமிக்கப்படுகிறது.

XRL A, direct

செயல் : (A) ← (A) (direct)

கட்டளையில் நேரடியாகக் குறிப்பிடப்பட்டுள்ள நினைவக இருப்பிடத்தில் உள்ள மதிப்பு, சேமக்கலத்தில் உள்ள மதிப்புடன் வாதியல் முறையில் XOR ஆகின்றது. இதன் முடிவு, சேமக்கலத்தில் சேமிக்கப்படுகிறது.

XRL A, @ Ri

செயல் : (A) ← (A) V ((Ri))

பதிவேடு Ri குறிப்பிடுகின்ற உள்அக RAM -இன் நினைவக இருப்பிடத்தில் உள்ள மதிப்பு, சேமக்கலத்தில் உள்ள மதிப்புடன் வாதியல் முறையில் XOR ஆகின்றது. இதன் முடிவு, சேமக்கலத்தில் சேமிக்கப்படுகிறது.

XRL A, # data

செயல் : (A) ← (A) ⊻ # data

கட்டளையின் இரண்டாவது பைட்டில் குறிப்பிடப்பட்டுள்ள மதிப்பு, சேமக்கலத்தில் உள்ள மதிப்புடன் வாதியல் முறையில் XOR ஆகின்றது. இதன் முடிவு, சேமக்கலத்தில் சேமிக்கப்படுகிறது.

XRL direct, A

செயல் : (direct) ← (direct) ⊻ (A)

சேமக்கலத்தில் உள்ள மதிப்பு, கட்டளையில் நேரடியாகக் குறிப்பிடப்பட்டுள்ள நினைவக இருப்பிடத்தில் உள்ள மதிப்புடன் வாதியல் XOR ஆகின்றது. இதன் முடிவு, அதே நினைவக இருப்பிடத்தில் சேமிக்கப்படுகிறது.

XRL direct, # data

செயல் : (direct) ← (direct) ⊻ #data

கட்டளையின் இரண்டாவது பைட்டில் குறிப்பிடப்பட்டுள்ள மதிப்பு கட்டளையில் நேரடியாகக் குறிப்பிடப்பட்டுள்ள நினைவக இருப்பிடத்தில் உள்ள மதிப்புடன் வாதியல் முறையில் XOR ஆகின்றது. இதன் முடிவு, அதே இருப்பிடத்தில் சேமிக்கப்படுகிறது.

CLR A செயல் : சேமக்கலத்தைக் காலியாக்கு :

பைட்டுகள் : 1 சுழற்சிகள் : 1

CLR A

செயல் : (A) ← 0

சேமக்கலம் காலியாகிறது. எந்தக் கொடிகளும் பாதிக்கப்படுவதில்லை.

CPL A செயல் : சேமக்கலத்தை எதிர்மறையாக்கு

பைட்டுகள் : 1 சுழற்சிகள் : 1

செயல் : (A) ← $\overline{(A)}$

சேமக்கலத்தில் உள்ள ஒவ்வொரு பிட்டும், வாதியல் முறையில் எதிர்மறையாகின்றது. எந்தக் கொடிகளும் பாதிக்கப்படுவதில்லை.

RL A

செயல் : சேமக்கலத்தை இடது புறமாகச் சுழற்று

பைட்டுகள் : 1 சுழற்சிகள் : 1

செயல் : (An+1) ← (An) n = 0-6

(A0) ← (A7)

சேமக்கலத்தில் உள்ள 8-பிட் மதிப்புகளும் இடது புறமாக ஒரு பிட் அளவு சுழற்றப்படுகின்றன. பிட் 1-இல் உள்ள மதிப்பானது பிட் 0-க்குச் செல்கின்றது. எந்தக் கொடிகளும் பாதிக்கப்படுவதில்லை.

RLC A பைட்டுகள் : 1 சுழற்சிகள் : 1

செயல் : சேமக்கலத்தைச் சுமைக் கொடியின் வழியாக இடது புறமாகச் சுழற்று.

செயல் : (An+1) ← (An) n = 0-6

(A0) ← (C)

(C) ← (A7)

RR A செயல் : சேமக்கலத்தை வலது புறமாகச் சுழற்று.

பைட்டுகள் : 1 சுழற்சிகள் : 1

செயல் : (An) ← (An+1) n= 0-6

(A7) ← (A0)

சேமக்கலத்தில் உள்ள 8-பிட்டுகளும் வலது புறமாக ஒரு பிட் அளவு சுழற்றபடுகின்றன. பிட் 0 ஆனது பிட் 1-க்குச் சுழற்சியின்போது செல்கிறது. எவ்விதக் கொடிகளும் பாதிக்கப்படுவதில்லை.

RRC A பைட்டுகள் : 1 சுழற்சிகள் : 1

செயல் : சேமக்கலத்தைச் சுமையுடன் வலது புறமாகச் சுழற்று.

(A) ← (An+1) n = 0-6

(A7) ← (C)

(C) ← (A0)

சேமக்கலத்தில் உள்ள 8 பிட்டுகளும் மற்றும் சுமைக் கொடியும் சேர்ந்து, வலதுபுறமாக ஒரு பிட் அளவு சுழலுகின்றன. பிட் 0 சுமைக் கொடிக்கும், சுமைக் கொடியில் ஏற்கனவே இருந்த மதிப்பானது பிட் 1-க்கும் செல்கின்றன. வேறு கொடிகள் பாதிக்கப்படுவதில்லை.

SWAP A

செயல் : சேமக்கலத்துடன் நிப்பில்களைப் பரிமாற்றுதல்

பைட்டுகள் : 1 சுழற்சிகள் : 1

$(A_{3-0}) \rightleftarrows (A_{7-4})$

இந்தக் கட்டளை, சேமக்கலத்தில் உள்ள தாழ் மற்றும் உயர் வரிசை நிப்பில்களின் மதிப்புகளை ஒன்றுக்கொன்று பரிமாற்றம் செய்து கொள்கின்றது. இது 4 பிட்டுகளை மட்டும் சுழற்றும் ஒரு கட்டளை ஆகும். எவ்விதக் கொடிகளும் பாதிக்கப்படுவதில்லை.

பூலியின் மாறிக் கையாடல் கட்டளைகள் :

CLR bit செயல் : பிட்டைக் காலியாக்கு பைட்டுகள் : 1 சுழற்சிகள் : 1

CLR C

(CY) - 0

செயல் : CLR

(bit) ← 0

சுமைக்கொடி காலியாகின்றது. வேறு கொடிகள் எதுவும் பாதிக்கப்படுவதில்லை.

SET B <bit> செயல் : பிட்டை அமைக்கவும்

பைட்டுகள் : 1/2 சுழற்சிகள் : 1

SET B C

செயல் : (C) ← 1

சுமைக் கொடி அமைவு நிலையை (1) அடைகிறது. வேறு கொடிகள் எதுவும் பாதிக்கப்படுவதில்லை.

SET B bit

செயல் : (bit) ← 1

கட்டளையில் நேரடியாகக் குறிப்பிடப்பட்டுள்ள முகவரிக்குரிய பிட், அமைவுநிலைக்குச் செல்கிறது. கொடிகள் எதுவும் பாதிக்கப்படுவதில்லை.

CPL bit செயல் : பிட்டை எதிர்மறையாக்கு

பைட்டுகள் : 1/2 சுழற்சிகள் : 1

CPL C

$(CY) \leftarrow \overline{(E)}$

சுமைக் கொடியில் உள்ள மதிப்பானது எதிர்மறையாகின்றது. வேறு எந்தக் கொடிகளும் பாதிக்கப்படுவதில்லை.

CPL bit

$(bit) \leftarrow \overline{(bit)}$

கட்டளையில் நேரடியாகக் குறிப்பிடப்பட்டுள்ள முகவரிக்குரிய பிட் எதிர்மறையாகிறது. வேறு கொடிகள் எதுவும் பாதிக்கப்படுவதில்லை.

ANL C, <Src - bit>

செயல் : பிட் மாறிகளுக்கான வாதியல் AND

பைட்டுகள் : 2 சுழற்சிகள் : 2

ANL C, bit

செயல் : $(C) \leftarrow (C) \wedge (bit)$

கட்டளைகளில் நேரடியாகக் குறிப்பிடப்பட்டுள்ள முகவரிக்குரிய பிட்டின் மதிப்பு சுமைக் கொடியில் உள்ள மதிப்புடன் வாதியல் முறையில் AND ஆகின்றது. இதன் முடிவு சுமைக் கொடியில் சேமிக்கப்படுகிறது. வேறு கொடிகள் எதுவும் பாதிக்கப்படுவதில்லை.

ANL C / bit

செயல் : $(C) \leftarrow (C) \wedge \overline{(bit)}$

கட்டளைகளில் குறிப்பிடப்பட்டுள்ள முகவரிக்குரிய பிட்டில் உள்ள மதிப்பின் எதிர்மறை, சுமைக் கொடியில் உள்ள மதிப்புடன் வாதியல்முறையில் AND ஆகின்றது. இதன் முடிவுக் சுமைக் கொடியில் சேமிக்கப்படுகின்றது. வேறு கொடிகள் எதுவும் பாதிக்கப்படுவதில்லை.

ORL C, <Src - bit> பைட்டுகள் : 2 சுழற்சிகள் : 2

செயல்: பிட் மாறிகளுக்கான வாதியல் OR

ORL C, bit

செயல் : (C) ⟵ (C) V (bit)

பூலியன் மதிப்பு 1 எனில். சுமைக் கொடியை அமைவு நிலையில் (1) வைக்கும். கட்டளையில் நேரடியாகக் குறிப்பிடப்பட்டுள்ள முகவரிக்குரிய பிட்டின் மதிப்பு, சுமைக் கொடியில் உள்ள மதிப்புடன் வாதியல் முறையில் OR ஆகின்றது. இதன் முடிவு, சுமைக் கொடியில் சேமிக்கப்படுகிறது. வேறு எந்தக் கொடிகளும் பாதிக்கப்படுவதில்லை.

ORL C, / bit

செயல் : (C) ⟵ (C) V (bit)

கட்டளையில் நேரடியாகக் குறிப்பிடப்பட்டுள்ள முகவரிக்குரிய பிட்டில் உள்ள மதிப்பின் எதிர்மறை, சுமைக் கொடியில் உள்ள மதிப்புடன் வாதியல் முறையில் OR ஆகின்றது. இதன் முடிவு, சுமைக் கொடியில் சேமிக்கப்படுகின்றது.

MOV <dest - bit>. <Src - bit>

செயல் : பிட் தரவை நகர்த்து பைட்டுகள் : 2 சுழற்சிகள் : 1/2

விளக்கம் : இரண்டாவது இயமியால் குறிப்பிடப்பட்ட பூலியன் மாறி, முதல் இயமியால் குறிப்பிடப்பட்ட இருப்பிடத்தில் வைக்கப்படும். ஒரு இயமி சுமைக் கொடி ஆகும். மற்றொன்று நேரடியாக முகவரியிடக்கூடிய பிட் ஆகும். வேறு எந்தப் பதிவேடும் அல்லது கொடிகளும் பாதிக்கப்படுவதில்லை.

MOV C, bit

செயல் : (C) ⟵ (bit)

திட்டநிரல் மற்றும் எந்திரக் கட்டுப்பாட்டுக் கட்டளைகள் :

ACALL addr II

செயல் : முழுமையான அழைப்பு பைட்டுகள் : 2 சுழற்சிகள் : 2

ACALL என்பது, இரண்டு பைட் நீளம் கொண்ட கட்டளை ஆகும். ACALL நிபந்தனை இல்லாமல், குறிப்பிடப்பட்ட முகவரி இருப்பிடத்திலுள்ள துணை நிரற்செயன்முறையை அழைக்கலாம். இந்தக் கட்டளை, PC-ஐ இரண்டு மடங்கு அதிகரிக்கச் செய்கிறது. பின்னர் 16-பிட், முடிவினை அடுக்கில் (தாழ்வரிசை பைட் முதலில்) தள்ளும். அடுக்குக் குறிமுள்ளை இருமடங்கு அதிகரிக்கச் செய்கிறது.

செயல் : ACALL

(PC) ⟵ (PC) + 2

(SP) ⟵ (SP) + 1

(SP) ⟵ (PC_{7-0})

(SP) ⟵ (SP) + 1

((SP)) ⟵ (PC_{15-8})

(PC_{10-0}) ⟵ பக்க முகவரி

இந்தக் கட்டளை PC-யின் மதிப்பை இரண்டு தடவை அதிகரிக்கச் செய்வதன் மூலம், அடுத்துள்ள கட்டளை இருக்கின்ற நினைவக முகவரியைப் பெறுகின்றது. அதன் பின் அந்த 16-பிட் முகவரியை அடுக்கு நினைவகத்திற்குத் (LSB முதலில்) தள்ளுகிறது மற்றும் அடுக்குக் குறிமுள்ளின் மதிப்பை, இரண்டு மடங்கு அதிகரிக்க செய்கின்றது. பின்னர் கட்டுப்பாடு எங்குச் செல்ல வேண்டுமோ அதன் இலக்கு முகவரி (target address) - ஐ பெறுவதற்காக, அதிகரிக்கப்பட்ட PC -யில் உள்ள உயர் வரிசை 5 பிட்டுகளும் செயற்பாடு குறியீடின் (opcode) 7-5 பிட்டுகளும் மற்றும் கட்டளையின் இரண்டாவது பைட்டும் ஒன்று சேருகின்றன.

இதில், CALL கட்டளையை அடுத்து வருகின்ற கட்டளை எந்தப் பக்கத்தில் உள்ளதோ அதே 2K பக்கத்திற்கு உள்ளாக மட்டுமே, கட்டுப்பாடு செல்ல முடியும். இந்தக் கட்டளைகளின்போது, எவ்விதக் கொடிகளும் பாதிக்கப்படுவதில்லை.

LCALL addr 16

செயல் : நீண்ட அழைப்பு பைட்டுகள் : 3 சுழற்சிகள் : 2

விளக்கம் : LCALL கட்டளை, அந்தக் கட்டளையில் குறிப்பிடப்பட்டுள்ள முகவரி இருப்பிடத்தில் இருக்கும் துணை நிரற்செயன்முறையை அழைக்கின்றது. இந்தக் கட்டளை. அடுத்த கட்டளையின் முகவரியை உருவாக்குவதற்கு PC-இன் மதிப்பை மூன்று மடங்கு கூட்டுகிறது. (அதிகரிக்கிறது). அதிகரிக்கப்பட்ட முகவரியில் மதிப்பு, அடுக்கு நினைவகத்தில் (LSB பைட் முதலில்) சேகரிக்கப்படுகிறது. அதன் பின்பு, அடுக்குக் குறிமுள் இரண்டு மடங்கு அதிகரிக்கிறது. இதில் துணை நிரற்செயன்முறை 64K பைட் கொண்ட நினைவக இருப்பிடத்தில், எங்கு வேண்டுமானாலும் இருக்கலாம். எந்த கொடிகளும், இந்தக் கட்டளையினால் பாதிக்கப்படுவதில்லை.

LCALL கட்டளை, மூன்று பைட் நீளம் கொண்டது. இந்தக் கட்டளை, துணை நிரற்செயன்முறைகளை அழைபபதற்கும் மற்றும் குறுக்கீடுகளுக்கான துணை நிரற்செயன்முறைகளை அழைப்பதறகும் பயன்படுகிறது.

செயல் : L CALL

$(PC) \leftarrow (PC) + 3$

$(SP) \leftarrow (SP) + 1$

$((SP)) \leftarrow (PC_{7\text{-}0})$

$(SP) \leftarrow (SP) + 1$

$((SP)) \leftarrow (PC_{15\text{-}8})$

$(PC) \leftarrow addr_{15\text{-}0}$

RET செயல் : துணை நிரற்செயன்முறையிலிருந்து திரும்புதல்

பைட்டுகள் : 1 சுழற்சிகள் : 2

REL கட்டளையின்மூலம் திட்டநிரல் எண்ணியின் உயர் மற்றும் தாழ் வரிசை பைட்டுகள், தொடர்ச்சியாக அடுக்கிலிருந்து எடுக்கப்படுகின்றன. முதலில் PC-இன் உயர்வரிசை பைட் அடுக்கிலிருந்து எடுக்கப்படுவதால், SP-இன் மதிப்பு ஒன்று குறைகின்றது. அடுத்ததாகத் தாழ் வரிசை பைட் அடுக்கிலிருந்து எடுக்கப்படுகின்றது. அதனால், SP-இன் மதிப்பு மேலும் ஒன்று குறைகின்றது. பொதுவாக, இந்தக் கட்டளை LCALL அல்லது ACALL கட்டளைகளைத் தொடர்ந்து வரும் கொடிகள் எதுவும், இதனால் பாதிக்கப்படுவதில்லை.

செயல் : RET

$(PC_{15-8}) \leftarrow ((SP))$

$(SP) \leftarrow (SP) - 1$

$(PC_{7-0}) \leftarrow ((SP))$

$(SP) \leftarrow (SP) - 1$

RET 1 செயல் : குறுக்கீடிலிருந்து திரும்பு பைட்டுகள் : 1 சுழற்சிகள் : 2

RET 1 கட்டளை செயல்படும்போது, PC-யின் உயர் மற்றும் தாழ் வரிசை பைட்டுகள் அடுக்கிலிருந்து எடுக்கப் (POP) படுகின்றபோது, குறுக்கீடு வாதியலை ஒப்படைக்கிறது. இதன்மூலம், ஏற்கனவே சேவை செய்யப்பட்ட குறுக்கீடுக்கு சமமான முந்துரிமை கொண்ட குறுக்கீடுகள் நிலுவையில் இருந்தால், அவை ஏற்றுக்கொள்ளப்படும். அடுக்குக் குறிமுள், இரண்டு மதிப்பு குறைகின்றது. வேறு எந்தப் பதிவேடுகளும் பாதிக்கப்படுவதில்லை. RET 1 கட்டளை மூலம் முதன்மைத் திட்டநிரலுக்குத் திரும்பிச் செல்கின்ற கட்டுப்பாடு 1 அது ஏற்கனவே எந்த கட்டளையைச் செயல்படுத்தி விட்டு வந்ததோ, அதற்கு அடுத்த கட்டளையை செயல்படுத்த ஆரம்பிக்கும். RET 1 கட்டளை செயல்படுகின்றபோது, ஒருவேளை அதே அளவு முந்துரிமை அல்லது அதைவிடக் குறைவான முந்துரிமை கொண்ட வேறு குறுக்கீடுகள் ஏதாவது நிலுவையில் இருந்தால், அந்தக் குறுக்கீடைச் செயல்படுத்துவதற்கு முன்பாக, ஒரே ஒரு கட்டளையை மட்டும் செயல்படுத்தும்.

செயல் : RET 1

$(PC_{15-8}) \leftarrow ((SP))$

$(SP) \leftarrow SP) - 1$

$(PC_{7-0}) \leftarrow ((SP))$

$(SP) \leftarrow (SP) - 1$

AJMP addr II செயல் : நிபந்தனையற்ற தாவல்

பைட்டுகள் : 2 சுழற்சிகள் : 2

விளக்கம் : AJMP திட்டநிரல், செயல்படுத்துதலைக் குறிப்பிடப்பட்டுள்ள முகவரிக்கு மாற்றுகிறது. PC-இன் மதிப்பு, இரண்டு மடங்கு அதிகரிக்கின்றது. அதன் பிறகு PC-இல் உள்ள உயர்வரிசை ஐந்து பிட்டுகளும், செயற்பாடு குறியீடின் 7-5 பிட்டுகளும் மற்றும் கட்டளையின் இரண்டாவது பைட்டுடன் சேர்ந்து, தாவ வேண்டிய இருப்பிடத்தைத் தீர்மானிக்கின்றன.

செயல் : AJMP

$(PC) \leftarrow (PC) + 2$

$(PC_{15-0}) \leftarrow$ பக்க முகவரி

LJMP addr 16 செயல் : நீண்ட தாவல் பைட்டுகள் : 3 சுழற்சிகள் : 2

விளக்கம் : குறிப்பிடப்பட்ட முகவரிக்கு, நிபந்தனையற்ற கிளையை LJMP உண்டாக்குகிறது. PC-யின் உயர்வரிசை மற்றும் தாழ்வரிசை பைட்டுகளுக்குக் கட்டளையின் இரண்டாவது மற்றும் மூன்றாவது பைட்டுகளில் குறிப்பிடப்பட்டுள்ள 16 பிட் முகவரி சுமையேற்றப்படுகிறது. இதன்மூலம் 6 கி பைட் அளவுகொண்ட திட்டநிரல் நினைவக முகவரி, இருப்பிடத்தில் எங்கு வேண்டுமானாலும் தாவ முடியும். கொடிகள் எதுவும் பாதிக்கப்படுவதில்லை.

செயல் : LJMP

$(PC) \leftarrow add_{15-0}$

SJMP rel செயல்: குறுகிய தாவல் பைட்டுகள்: 2 சுழற்சிகள் : 2

விளக்கம்: முதலில் திட்டநிரல் எண்ணி இரண்டு மடங்கு அதிகரிக்கின்றது. அதன் பின்பு, கட்டளையின் இரண்டாவது பைட்டில் கூறப்பட்டுள்ள சார்பு முகவரி PC-இல் உள்ள மதிப்புடன் சேர்க்கப்பட்டுத் தாவ வேண்டிய இருப்பிடத்தைத் தேர்வு செய்கின்றது. இதன்மூலம் தாவ வேண்டிய இருப்பிடம், இந்தக் கட்டளையை அடுத்து வருகின்ற கட்டளையிலிருந்து முன் திசையில் 127 பைட் இருப்பிடங்களுக்கு உள்ளாகவும் மற்றும் பின் திசையில் 128 பைட் இருப்பிடங்களுக்கு உள்ளாகவும் இருக்கும்.

செயல்: SJMP

(PC) ← (PC) + 2

(PC) ← (PC) + rel

JMP @A + DPTR செயல்: மறைமுகத் தாவல் பைட்டுகள் : 1 சுழற்சிகள்: 2

விளக்கம் : சேமக்கலத்தில் உள்ள குறியிடப்படாத எட்டு பிட் மதிப்புடன் DPTR-இல் உள்ள 16 பிட் மதிப்பானது கூட்டப்பட்டு PC -இல் சுமையேற்றப்படும். இதுதான் அடுத்ததாக வருகின்ற கட்டளைத் தாழிடல்களுக்கு முகவரியாகும். சேமக்கலத்தின் மதிப்போ அல்லது DPTR-இன் மதிப்போ மாற்றம் அடையாது. கொடிகள் எதுவும் பாதிக்கப்படுவதில்லை.

செயல் : (P) ← (A) + (DPTR)

JZ rel செயல்: சேமக்கலம் O எனில், தாவு பைட்டுகள் : 2 சுழற்சிகள் : 2

விளக்கம் : சேமக்கலத்தில் உள்ள அனைத்துப் பிட்டுகளும் O எனில், சுட்டிக்காட்டப்பட்ட முகவரிக்குக் கிளையாகப் பிரியும்; இல்லையெனில், அடுத்த கட்டளையைத் தொடர்ந்து செய்யும். சேமக்கலத்தில் உள்ள மதிப்பு மாற்றப்படுவதில்லை. கொடிகள் எதுவும் பாதிக்கப்படுவதில்லை.

செயல்: (PC) ← (PC) + 2

IF (A) = 0

THEN (PC) ← (PC) + rel

JNZ rel செயல்: சேமக்கலம் O இல்லையெனில் தாவு,

பைட்டுகள் : 2 சுழற்சிகள் : 2

விளக்கம்: சேமக்கலத்தின் ஏதாவதொரு பிட் O எனில், குறிப்பிட்ட முகவரிக்குக் கிளையாகப் பிரியும். இல்லையெனில், அடுத்த கட்டளையைத் தொடர்ந்து செயல்படுத்தும். சேமக்கலத்தில் உள்ள மதிப்பு மாற்றப்படுவதில்லை மற்றும் கொடிகள் எதுவும் பாதிக்கப்படுவதில்லை.

செயல்: (PC) ← (PC) + 2

IF (A) $\neq$ 0

THEN (PC) ← (PC) + rel

JC rel செயல் : சுமை 1 ஆக இருந்தால் தாவு பைட்டுகள் : 2 சுழற்சிகள் : 2

விளக்கம் : சுமைக்கொடி அமைவு நிலையில் (1) இருந்தால், குறிப்பிட்ட முகவரிக்குக் கிளையாகப் பிரியும். இல்லையெனில், அந்தக் கட்டளைக்கு அடுத்துள்ள கட்டளையைத் தொடர்ந்து செயல்படுத்தும். எந்தக் கொடிகளும் பாதிக்கப் படுவதில்லை.

செயல்: (PC) ⟵ (PC) + 2
IF (PC) = 1
THEN
(PC) ⟵ (PC) + rel

JNC rel செயல்: சுமைக் கொடி 0 இல்லையெனில் தாவு,

பைட்டுகள் : 2 சுழற்சிகள் : 2

விளக்கம்: சுமை கொடி 0 இல்லையெனில், கட்டுப்பாடானது குறிப்பிட்ட முகவரிக்குத் தாவும். மாறாகச் சுமைக் கொடியின் மதிப்பு 1 எனில், அடுத்துள்ள கட்டளையில் இருந்து செயல்படுத்தும். சுமைக் கொடியின் மதிப்பு மாறுவதில்லை.

செயல் : (PC) ⟵ (PC) + 2
IF (C) = 0
THEN (PC) ⟵ (PC) + rel

JB bit, rel செயல்: பிட் இருப்பிடம் 1 எனில் தாவு பைட்டுகள் : 3 சுழற்சிகள் : 2

விளக்கம்: JB கட்டளையில் நேரடியாக முகவரியிடப்பட்டுள்ள பிட் இருப்பிடத்தில் உள்ள மதிப்பு, அமைவு நிலையில் (1) இருந்தால் மட்டும் கட்டுப்பாடு தாவும். மாறாகப் பிட்டின் மதிப்பு '0' எனில், அதற்கு அடுத்துள்ள கட்டளையில் இருந்து செயல்படுத்தும். பிட்டில் உள்ள மதிப்பு மாறுவதில்லை. கொடிகள் எதுவும் பாதிக்கப்படுவதில்லை.

செயல்: (PC) ⟵ (PC) + 3
IF (bit) = 1
THEN (PC) ⟵ (PC) + rel

JNB bit, rel செயல்: பிட் இருப்பிடம் 0 எனில் தாவு பைட்டுகள் : 3 சுழற்சிகள் : 2

விளக்கம்: JNB கட்டளையில் நேரடியாக முகவரியிடப்பட்டுள்ள பிட் இருப்பிடத்தில் உள்ள மதிப்பு 0 நிலையில் இருந்தால் மட்டுமே, கட்டுப்பாடு தாவும். மாறாகப் பிட்டின் மதிப்பு '1' எனில், அதற்கு அடுத்துள்ள கட்டளையில் இருந்து செயல்படுத்தும். பிட்டில் உள்ள மதிப்பு மாறுவதில்லை. கொடிகள் எதுவும் பாதிக்கப்படுவதில்லை.

செயல்: (PC) ⟵ (PC) + 3
IF (bit) = 0
THEN (PC) ⟵ (PC) + rel

JBC bit, rel செயல்: பிட் இருப்பிடம் '1' எனில் தாவும். பின்னர் பிட் '0' ஆகிவிடும்.
பைட்டுகள் : 3 சுழற்சிகள் : 2

விளக்கம் : JBC கட்டளையில் நேரடியாக முகவரி செய்யப்பட்டுள்ள பிட் இருப்பிடத்தில் உள்ள மதிப்பு, அமைவு நிலையில் (1) இருந்தால் மட்டுமே கட்டுப்பாடு தாவும். மேலும் அதன் மதிப்பு சுழி? (O) விடும். மாறாகப் பிட்டின் மதிப்பு '0' எனில், அதன் மதிப்பு மாறுவதில்லை. மேலும், அதற்கு அடுத்துள்ள கட்டளையில் இருந்து தொடர்ந்து செயல்படுத்தும். கொடிகள் எதுவும் பாதிக்கப்படுவதில்லை.

செயல்: (PC) ← (PC) + 3

IF (bit) = 1

THEN

(bit) ← 0

(PC) ← (PC) + rel

CJNE <delt - byte>, <SCR - byte>, rel பைட்டுகள் : 3 சுழற்சிகள் : 2

(உ-ம்) : CJNE A, direct, rel

செயல்: (PC) ← (PC) + 3

IF (A) <> (direct)

THEN

(PC) ← (PC) + relative offset

IF(A) < (direct)

THEN

(C) ← 1

ELSE

(C) ← 0

இந்தக் கட்டளை, நேரடி பைட் இருப்பிடத்தில் உள்ள மதிப்பைச் சேமக்கலத்தில் உள்ள மதிப்புடன் ஒப்பிடுகின்றது. இரண்டின் மதிப்புகளும் சமமாக இல்லையென்றால் மட்டுமே, கட்டுப்பாடு இலக்கு முகவரிக்குத் தாவும். இல்லையெனில், தாவாமல் அடுத்துள்ள கட்டளையைச் செயல்படுத்தும். கட்டளையில் கூறப்பட்டுள்ள நேரடி பைட்டில் உள்ள இருப்பிடத்தின் மதிப்பு, சேமக்கலத்தில் உள்ள மதிப்பைவிட அதிகமாக இருந்தால், சுமைக் கொடி அமைவு நிலையை (1) அடையும். மாறாக குறைவாக இருந்தால், சுமை கொடி சுழி? (O) விடும்.

CJNE A, #data, rel (Compare immediate to ACC, and Jump if not equal)

செயல்: (PC) ← (PC) + 3

IF (A) <> data

THEN

(PC) ← (PC) + relative offset

IF(A) < data

THEN

(C) ← 1

ELSE

(C) ← 0

இந்தக் கட்டளை உடனடித் தரவைச் சேமக்கலத்தில் உள்ள மதிப்புடன் ஒப்பிடுகின்றது. இரண்டு மதிப்புகளும் சமமாக இல்லையென்றால், கட்டுப்பாடு இலக்கு முகவரிக்குத் தாவும். இல்லையெனில் தாவாமல் அடுத்துள்ள கட்டளையைச் செயல்படுத்தும். உடனடித் தரவின் மதிப்பு சேமக்கலத்தில் உள்ள மதிப்பை விட அதிகமாக இருந்தால், சுமைக் கொடி அமைவு நிலையை (1) அடையும். மாறாக குறைவாக இருந்தால், சுமைக் கொடி சுழியாகி (0) விடும்.

CJNE A, #data, rel (Compare immediate to register, and Jump if not equal)

செயல்: (PC) ← (PC) + 3
IF (Rn) <> data
THEN
(PC) ← (PC) + relative offset
IF(Rn) < data
THEN
(C) ← 1
ELSE
(C) ← 0

இந்தக் கட்டளை உடனடித் தரவைப் பதிவேட்டில் உள்ள மதிப்புடன் ஒப்பிடுகின்றது. இரண்டு மதிப்புகளும் சமமாக இல்லையென்றால், கட்டுப்பாடு இலக்கு முகவரிக்குத் தாவும். மாறாக சமமாக இருந்தால், அதற்கு அடுத்த கட்டளையைச் செயல்படுத்தும்.

உடனடித் தரவின் மதிப்பானது, பதிவேடு Rn இல் உள்ள மதிப்பை விட அதிகமாக இருந்தால், சுமைக் கொடி அமைவு நிலைக்குச் (1) செல்லும். மாறாகக், குறைவாக இருந்தால், சுழியாகி (0) விடும்.

CJNE @ Ri, #data, rel (Compare immediate to indirect, and Jump if not equal)

செயல்: (PC) ← (PC) + 3
IF ((Ri)) <> data
THEN
(PC) ← (PC) + relative offset
IF((Ri)) < data
THEN
(C) ← 1
ELSE
(C) ← 0

இந்தக் கட்டளை, பதிவேடு Ri குறிப்பிடுகின்ற உள்அக RAM நினைவக இருப்பிடங்களில் உள்ள மதிப்பினை, மறைமுகத் தரவு மதிப்புடன் ஒப்பிடுகின்றது. இரண்டு மதிப்புகளும் சமமாக இல்லையென்றால், கட்டுப்பாடு இலக்கு முகவரிக்குத் தாவும். மாறாகச் சமமாக இருந்தால், தாவாமல் அடுத்துள்ள கட்டளையைச் செயல்படுத்தும். மறைமுகத் தரவின் மதிப்பு, பதிவேடு Ri குறிப்பிடுகின்ற உள்அக RAM இருப்பிடத்தில் உள்ள மதிப்பைவிடக் குறைவாக இருந்தால், சுமைக்கொடி அமைவு நிலையை (1) அடையும். மாறாக அதிகமாக இருந்தால் சுமைக் கொடி சுழியாகி (0) விடும்.

DJNZ <byte>, (ret addr> பைட்டுகள் : 2/3 சுழற்சிகள் : 2
DJNZ Rn, rel

செயல்: (PC) ← (PC) + 2

(Rn) ← (Rn) + 1

IF(Rn) > 0 or (Rn) < 0

THEN

(PC) ← (PC) + rel

பதிவேடு Rn-இல் உள்ள மதிப்பு ஒன்று குறைகின்றது. தற்போது கிடைக்கப்பெறுகின்ற மதிப்பு '0' ஆக இல்லையெனில், கட்டுப்பாடு இலக்கு முகவரிக்குத் தாவும். மாறாக '0' ஆக இருந்தால், அடுத்துள்ள கட்டளையைச் செயல்படுத்தும். எந்தக் கொடிகளும் பாதிக்கப்படுவதில்லை.

DJNZ direct, rel

செயல்: (PC) ← (PC) + 2

(direct) ← (direct) - 1

IF (direct) > 0 or (Rn) < 0

THEN

(PC) ← (PC) + rel

நேரடி பைட் இருப்பிடத்தில் உள்ள மதிப்பு ஒன்று குறைகின்றது. தற்போது கிடைக்கப்பெறுகின்ற மதிப்பு '0' ஆக இல்லையெனில், கட்டுப்பாடு இலக்கு முகவரிக்குத் தாவும். மாறாக '0' ஆக இருந்தால், அடுத்துள்ள கட்டளையைச் செயல்படுத்தும். கொடிகள் எதுவும் பாதிக்கப்படுவதில்லை.

NOP செயல் : செயற்பாடு இல்லை பைட்டுகள் :1 சுழற்சிகள் : 2

விளக்கம் : கீழே கொடுக்கப்பட்ட கட்டளையில் இருந்து, செயல்பாடு தொடரும். PC தவிர பதிவேடு எதுவும் மாறுவதில்லை. அல்லது கொடிகள் பாதிக்கப்படுவதில்லை.

செயல் : (PC) (PC) + 1

15.20. 8051 திட்டநிரல்கள் :

1. சேமக்கலம் A, DPH மற்றும் DPL ஆகியவற்றை 30H என்ற மதிப்பைக் கொண்டு சுமையேற்றம் செய்வதற்கான திட்டநிரல்.

MOV A, # 30H	: பதிவேடு A – ஐ 30H கொண்டு சுமையேற்று.
MOV DPH, A	: (DPH) ← (A)
MOV DPL, A	: (DPL) ← (A)

2. இரண்டு 8 பிட் எண்களைக் கூட்டல்

MOV A, # 30H : (A) ← 30

MOV A, # 50H : (A) ← (A) + 50H

3. இரண்டு 16 பிட் எண்களின் கூடுதல் கணக்கிடல்

MOV DPTR, # 2040H	. (DPTR) ← 2040H (16 பிட் எண்)
MOV A, # 2BH	. (A) ← 2BH (இரண்டாவது 16 பிட் எண்ணின் தாழ்வரிசை பைட்) .
MOV B, # 20H	: (B) ← 20H (இரண்டாவது 16 பிட் எண்ணின் உயர்வரிசை பைட்)
ADD A, DPL	: தாழ்வரிசை பைட்டுகளைக் கூட்டு
MOV DPL, A	: தாழ்வரிசை பைட்டுகளின் கூடுதல் முடிவைக் காப்பாற்று.

MOV A, B	:	இரண்டாவது எண்ணின் உயர்வரிசை பைட்டை A-இல் வாங்கு.
ADDC A, DPH	:	உயர்வரிசை பைட்டுகளை தாழ்வரிசை பைட்டின் கூட்டலில் வரும் சுமை உடன் கூட்டு.
MOV DPH, A	:	உயர்வரிசை பைட்டின் கூட்டலின் முடிவைக் காப்பாற்று.

4. R0-இல் உள்ள ஒரு எண்ணின் 2வது நிரப்பியைக் கண்டுபிடித்தல்.

MOV A, R0	:	(A) ⟵ (R0)
CPL A	:	A-இன் 1ஆவது நிரப்பி
ADD A, # 01	:	2வது நிரப்பியைப் பெற 1-ஐ இதனுடன் கூட்டு

5. இரண்டு 8 பிட் எண்களின் கழித்தல்

MOV A, # 9F	:	முதல் எண்ணை A-இல் வாங்கு.
MOV R0, # 40	:	இரண்டாவது எண்ணை R0-இல் வாங்கு.
CLR C	:	சுமையைக் சுழியாக்கு.
SUBB A, R0	:	A - A - (R0)
SWAP A	:	இலக்கங்களை மாற்று.

6. இரண்டு 8 பிட் எண்களின் வகுத்தல்

MOV A, # 90	:	முதல் எண்ணை A-இல் வாங்கு.
MOV B, # 20	:	இரண்டாவது எண்ணை A-இல் வாங்கு.
DIV AB	:	A ÷ Bஇல் மீதி மற்றும் A-இல் ஈவு.

7. இரண்டு 8 பிட் எண்களின் பெருக்கல்

MOV A, # 8F	:	முதல் எண்ணை A- இல் வாங்கு
MOV B, # 79	:	இரண்டாவது எண்ணை B-இல் வாங்கு.
MUL AB	:	AXB, B-இல் முடிவின் உயர்வரிசை பைட் மற்றும் A-இல் முடிவின் தாழ்வரிசை பைட்.

8. 8 பிட் எண்களில் பத்தில் பெரும எண்ணைக் காணல்.

	MOV DPTR, # 2000	:	எண்களை நினைவகத்தில் சேமிப்பதற்குக் குறிமுள்ளை ஆரம்ப நிலையில் வை.
	MOV R0, # OA	:	எண்ணியைத் தொடக்க நிலையில் வைத்தல்
	MOV R3, # 00	:	பெருமம் = O
AGAIN :	MOVX A, @ DPTR	:	நினைவகத்தில் இருந்து எண்ணை வாங்கு
	CJNE A, R3, NE	:	பெரும எண்ணுடன் எண்ணை ஒப்பிடு
	AJMP SKIP	:	சமமாக இருந்தால் SKIP-க்குச் செல்
NE :	JC SKIP	:	சமம் இல்லையெனில் சுமையைச் சரி பார் சுமை இருந்தால் SKIP-க்குச் செல்.
	MOV R3, A	:	மாறாகப், பெருமம் = எண்
SKIP :	INC DPTR	:	நினைவகக் குறிமுள்ளை உயர்த்து
	DJNZ RD, AGAIN	:	எண்ணிக்கை = O எனில், எண்ணிக்கையைத் தாழ்த்து. நிறுத்து அல்லது AGAIN-க்குச் செல்.

9. பத்து 8 பிட் எண்களை ஏறு வரிசையில் அடுக்கு.

	MOV R0, # 09	·	எண்ணி 1-ஐ தொடக்க நிலையில் வை.
AGAIN:	MOV DPTR # 2000H	·	நினைவகக் குறிமுள்ளைத் தொடக்க நிலையில் வை.
	MOV R1, # 09	:	எண்ணி 2-ஐ தொடக்க நிலையில் வை.

BACK : MOV R2, DPL : நினைவக முகவரியின் தாழ்வரிசை பைட்டைக் காப்பாற்று.
MOVX A, @ DPTR : எண்ணை வாங்கு
MOV B, A : எண்ணைக் காப்பாற்று
INC DPTR : நினைவக குறிமுள்ளை உயர்த்து
MOVX A, @ DPTR : அடுத்த எண்ணை வாங்கு
CJNE A, B, NE : சமமாக இல்லையெனில் பெரும அல்லது குறைந்த எண்ணைச் சரி பார்.
AJMP SKIP : இல்லையேல் SKIP-க்குச் செல்.
NE : JNC SKIP . IF
MOV DPL, R2 : (இரண்டு நினைவக இருப்பிடங்களில் உள்ளவற்றை
MOV X @ DPTR, A மாற்று).
INC DPTR :
MOV A, B :
MOV X @ DPTR, A :

SKIP DJNZ R1, BACK : R1 ≠ 0 எனில், BACK-க்குச் செல்.
DJNZ R0, AGAIN : R0 ≠ 0 எனில், AGAIN-க்குச் செல்.

10. அடுக்கில் சேமிக்கப்பட்டுள்ள 10 எண்களின் கூட்டுத்தொகை.

MOV DPTR, # 2200H : நினைவகக் குறிமுள்ளைத் தொடக்க நிலையில் வை.
MOVX A, @ DPTR : எண்ணிக்கையை வாங்கு.
MOV R 0, # 10 : இரட்டுறுத்தல் எண்ணித் தொடக்க நிலையில் வை.
INC DPTR : அடுக்கிலுள்ள எண்ணிகளுக்குக் குறிமுள்ளைத் தொடக்க நிலையில் வை.
MOV R1, ≠ 00 : முடிவு = O
BACK: MOVX A, @ DPTR : எண்ணை வாங்கு
ADD A, R1 : A ← முடிவு + A
MOV R1, A : முடிவு ← A
INC DPTR : அடுக்குக் குறிமுள்ளை உயர்த்து
DJNZ R0, BACK : இரட்டுறுத்தல் எண்ணியைக் குறை, O இல்லையெனில் திரும்பச் செல்.
MOV DPTR, # 2300H : நினைவகக் குறிமுள்ளைத் தொடக்க நிலையில் வை.
MOV A, R1 : முடிவை வாங்கு.
MOV X @ DPTR, A : முடிவைச் சேமி

உதாரணம் : 2200H = 04H
2201H = 20H
2202H = 15H
2203H = 13H
2204H = 22H
முடிவு = 20 + 15+ 13+ 22 = 6AH

முடிவு 2300H = 6AH

வினாக்கள் :

1. நுண்கட்டுப்படுத்தி என்றால் என்ன?
2. 8051 நுண்கட்டுப்படுத்தியில் உள்ள I/O துறைமுகங்கள் எத்தனை? விளக்குக.
3. நுண்கட்டுப்படுத்தியின் பயன்களைக் கூறுக.
4. திட்டநிரல் எண்ணி மற்றும் தரவுக் குறிமுள் பற்றி விளக்குக.
5. 8051-இல் உள்ள ALU பற்றி விளக்குக.
6. 8051 நுண்கட்டுப்படுத்தியின் முனைப் படம் வரைந்து விளக்குக.
7. புறக் குறுக்கீடுகளைப் பற்றி விளக்குக?
8. 8051-இன் கட்ட வரைபடம் வரைந்து அதனை விளக்குக?
9. சிறப்புச் செயல்சார் பதிவேடுகளை வரிசைப்படுத்தி அவற்றின் செயல்களை விளக்குக.
10. உள்அக RAM -இன் அமைப்பைப் படத்துடன் விவரி
11. 8051 நுண்கட்டுப்படுத்தியின் நினைவக அமைப்பைச் சுருக்கமாக விளக்குக.
12. புறத் தரவு நினைவகம் மற்றும் புறத் திட்டநிரல் நினைவகம் பற்றி விளக்குக.
13. PSW பதிவேடு மற்றும் SMOD பதிவேடுகளை விவரி.
14. வரைபடத்துடன், நுண்கட்டுப்படுத்தியின் I/O துறைமுகங்களை விளக்குக.
15. TMOD மற்றும் TCON பதிவேடுகளை விவரி
16. 8051இல் உள்ள முகவரியிடல் வகைகளைச் சான்றுகளுடன் விளக்குக.
17. MOVX மற்றும் MOVC கட்டளைகளை உதாரணங்களுடன் விளக்குக.
18. CALL மற்றும் RETURN கட்டளைகளை விளக்குக.
19. 8051-இல் பயன்படுத்தக்கூடிய ஒப்பிடு கட்டளைகளைச் சான்றுகளுடன் விளக்குக.
20. INC, ADD மற்றும் ADDC கட்டளைகள் எவ்வாறு 8051-இல் பயன்படுத்தப்படுகின்றன என்பதை விளக்குக.

கலைச் சொற்கள்

Accumulator	- சேமக்கலம்
Address bus	- முகவரி ஊர்தி
Address Latch Enable	- முகவரி தாழ் செயலியக்கச் சைகை
Architecture	- கட்டமைப்பு
Arithmetic and Logical Unit	- கணித மற்றும் வாதியல் அலகு
Arithmetic Group	- கணிதத் தொகுப்பு
Ascending Order	- ஏறு வரிசை
ASCII Code	- அமெரிக்கச் செய்திப் பரிமாற்றப் படித்தரக் குறியீடு
Assembly Language	- இடைநிலை மொழி
Assembler	- கூட்டாளர்
Asynchronous Transmission Mode	- இசைவற்ற செலுத்தீட்டு முறை
Auxillary Carry	- துணைச் சுமை
Base	- அடித்தளம்
Bidirection	- இருதிசை
Binary coded decimal	- இருமக் குறியீட்டுப் பதின்மம்
Branch group	- கிளைத் தொகுப்பு
Break	- இடைவெளி
Buffer	- தாங்கி
Bus	- ஊர்தி

Call Instructions	- அழைப்புக் கட்டளைகள்
Carry	- சுமை
Carrier frequency	- ஏந்தி அதிர்வெண்
Central Processing Unit	- மையச் செயலகம்; நடுவண் செயலாக்க அலகு
Channel	- தடங்கள்
Chip	- சில்
Clock cycle	- கடிகாரச் சுற்று
Code	- குறியீடு
Command	- கட்டளை
Compare	- ஒப்பிடு
Control bus	- கடடுப்பாட்டு ஊர்தி
Control group	- கட்டுப்பாட்டுத் தொகுப்பு
Data	- தரவுகள்
Data bus	- தரவு ஊர்தி
Data transfer group	- தரவு மற்றும் தொகுப்பு
D/A converter	- எண்ணிலக்க ஒப்புமை மாற்றி
Decoder	- குறியீடு மாற்றி; குறியீடு பெயர்ப்பி
Descending order	- இறங்கு வரிசை
Destination	- சேர்ப்பிடம்
Digital Computer	- எண்ணிலக்க கணினி
Displacement address	- இடப்பெயர்ச்சி முகவரி

Direct addressing	- நேரடி முகவரியிடல்
Direction	- திசை
Directives	- வழிமுறைகள்
Disable	- செயல்முடக்கம்
Dividend	- வகுபடும் எண்
Divisor	- வகுக்கும் எண்
Effective address	- செயலூக்கமுடைய முகவரி
Enable	- செயலியக்கம்
Error	- பிழை
External	- புறம்
External access pin	- புற அடை முனை
Extra segment	- கூடுதல் பகுதி
Flag	- கொடி
Fetch	- கொணர்
File	- கோப்பு
Flip flop	- நிலைமாற்றி
Floppy disk	- நெகிழ்வட்டு
Frequency	- அதிர்வெண்
Generator	- மின்னாக்கி
General purpose registers	- சிறப்பு நோக்கப் பதிவேடுகள்
Hard disk	- வன்வட்டு
Hard ware	- வன்பொருள்

Hardware Triggered Mode	-	வன்பொருள் தொடக்கி முறை
Immediate addressing	-	உடனடி முகவரியிடல்
Implicit addressing	-	உள்ளடக்க முகவரியிடல்
Index	-	சுட்டிக்காட்டி
Input devise	-	உள்ளீட்டகம்
Instruction	-	கட்டளை
Instruction pointer	-	கட்டளைச் சுட்டிக்காட்டி
Instruction register	-	கட்டளைப் பதிவேடு
Instruction template	-	கட்டளை மாதிரிகள்
Interfacing	-	இடைப் பரிமாறல்
Internal	-	உள்ளார்ந்த ; உள் அகம்
Interrupt	-	குறுக்கீடு ; தடங்கல்
I/O port	-	உள்ளீடு/வெளியீடு நிலையம் உள்ளீடு/வெளியீடு துறைமுகம்
Keyboard	-	விசைப் பலகை
Latch	-	தாழிடல்
Least Significant Bits	-	சிறும முக்கியத்துவம் வாய்ந்த பிட்(
Location	-	இருப்பிடம்
Logical Group	-	வாதியல் தொகுப்பு
Loop	-	வளையம்
Look-up table	-	பார்வை அட்டவணை ; நோக்கீட்டு அணிப் பட்டிகள்

Machine cycle - கருவிச் சுற்று

Machine control group - கருவிக் கட்டுப்பாட்டுத் தொகுப்பு

Machine language - இயந்திர மொழி

Magnetic memory - காந்த நினைவகம்

Magnetic reader - காந்தப் படிப்பான்

Magnetic tape - காந்த நாடா

Mask able - மறைக்கக் கூடிய

Maximum - அதிக பட்சம்

Memory - நினைவகம்

Memory location - நினைவக இருப்பிடம்

Memory mapped - நினைவக படமம்

Micro Computer - நுண் கணினி

Micro Controller - நுண் கட்டுப்படுத்தி

Micro Processor - நுண்முறைப்படுத்தி

Minimum - குறைந்தபட்சம்

Mnemonics - செயற்குறிப்பு

Mode - முறை

Mode set register - முறை அமைப்புப் பதிவேடு

Monitor - திரையகம்

Most significant bits - பெரும முக்கியத்துவம் வாய்ந்த பிட்டுக
ள்

Multi processor - பன்முறைப்படுத்தி

Multiplexing	- பலவின் ஒன்றாக்குதல்
Opcode	- கருவிக் குறியீடு ; இயக்கக் குறியீடு
Open drain	- திறந்த வடிகால்
Operand	- இயமிகள்
Organization	- அமைப்பு
Oscillator	- அலையியற்றி
Output device	- வெளியீட்டகம்
Over flow	- பிரவாகம் ; பொங்கி வழி
Over ride	- புறக்கணிப்பு
Parity	- சமநிலை ; ஒப்புமை
Peripheral devices	- புறவியலான சாதனங்கள்
Physical address	- தரவு சாார்ந்த முகவரி
Physical devices	- உள்கட்டச் சாதனங்கள்
Pin	- மின்முனை
Pin configuration	- முனை அமைப்பு
Port	- நிலையம் ; துறைமுகம்
Priority level structure	- முன்னுரிமை மட்ட அமைப்பு
Programe	- திட்டநிரல்
Program counter	- திட்டநிரல் எண்ணி
Program status word	- திட்டநிரல் நிலைப்பாடு வார்த்தை ; திட்நிரல் நிலைப்பாடு சொல்

DR. T.P.M. LIBRARY

Printer - அச்சுப் பொறி

Pulse - துடிப்பு

Random Access Memory (RAM) - நோக்கமின்றி அணுகும் நினைவகம்

Rate Generator - தகைவேக மின்னாக்கி

Read Only Memory (ROM) - படித்தல் மட்டும் நினைவகம்

Register - பதிவேடு

Register addressing - பதிவேடு முகவரியிடல்

Register Bank - பதிவேடு வைப்பகம்

Register Indirect addressing - பதிவேடு மறைமுக முகவரியிடல்

Remainder - மீதி

Reset - மறு அமைவு

Return instructions - திருப்புக் கட்டளைகள்

Rotate - சுழற்சி

Scanner - வருடி

Segment - பகுதி

Segment override prefix - பகுதிப் புறக்கணிப்பு முற்சேர்க்கை

Semi conductor memory - குறைக்கடத்தி நினைவகம்

Serial data mode set - தொடர்வரிசை தரவு

Set - அமைவு

Service - சேவை

Sign - குறியீடு

Signal - சைகை

Software - மென்பொருள்

Special purpose register - சிறப்பு நோக்கப் பதிவேடு

Square wave - சதுரலை

Stack pointer - அடுக்குச் சுட்டிக்காட்டி ; அடுக்குக் குறிமுள்

Stack register - அடுக்குப் பதிவேடு

Stack top - அடுக்கு உச்சி

String - கம்பி

Strobe - திரிபலை

Subroutine - துணைநிரற் செயல்முறை

Synchronous Transmission Mode - ஒத்தியங்கு செலுத்தீட்டு முறை

Terminal - முனையகம்

Timing and Control Unit - நேர மற்றும் கட்டுப்பாட்டு அலகு

Timing Signals - நேர சைகைகள்

Timer - நேரக்காட்டி ; நேரச்சுற்று

Trigger level - தொடக்கி மட்டங்கள்

Variable - மாறி

Vectored Interrupt - திசையன் குறுக்கீடு

2's Complement - இரண்டின் நிரப்பி

DR. T.P.M. LIBRARY ★ MADURAI KAMARAJ UNIVERSITY ★
299067