विश्वविद्यालय अनुदान आयोग
University Grants Commission
Quality higher education for all

माइक्रोप्रोसेसर माइक्रोकंट्रोलर

एक परिचय

लेखक
प्रो. आशीष वर्मा
रितु आर्य

लेखक

प्रो. आशीष वर्मा

रितु आर्य

लेखकीय

आज के इस कंप्यूटर युग में कंप्यूटर का उपयोग इतना बढ़ गया है कि हम इसके अंदर की प्रणाली को देखना ही नहीं चाहते हैं। परन्तु यह जानना आवश्यक है कि कंप्यूटर के अंदर की प्रणालियाँ किस तरह से काम करती हैं। जैसा कि हम जानते हैं कि कंप्यूटर का ह्रदय उसका माइक्रोप्रोसेसर है, इस किताब में माइक्रोप्रोसेसर और उससे जुडी संदेश को विस्तार से समझाया गया है।

इस किताब के पहले अध्यायों में माइक्रोप्रोसेसर 8085 को विशेष रूप से विस्तारपूर्वक समझाया गया है। यह मेरा मानना है की 8085 माइक्रोप्रोसेसर को समझने के बाद अन्य कोई माइक्रोप्रोसेसर समझा जा सकता है।

इस किताब के सातवें अध्याय में 6800 माइक्रोप्रोसेसर को समझाया है जो एक 16 बिट का है। आठवें अध्याय में पेंटियम और पेंटियम प्रो माइक्रोप्रोसेसर के अंकगणितीय और लॉजिकल प्रचालन को समझाया गया है।

यह किताब उन सभी छात्रों / छात्राओं के लिए लाभप्रद होगी जो स्नातक और स्नातकोत्तर पर इस विषय को पढ़ रहे हैं ।

हम आभारी है इस विश्वविद्यालय की कुलपति प्रो.नीलिमा गुप्ता मैडम का जिन्होंने हमें मौका दिया इस किताब के प्रकाशन का और हमे प्रोत्साहित किया इस उत्कृष्ट कार्य को करने के लिए।

आशीष वर्मा
रितु आर्या

अनुक्रमाणिका

अध्याय -1

1.1 माइक्रोप्रोसेसर क्या है

1.2 माइक्रोप्रोसेसर का इतिहास

1.3 माइक्रोप्रोसेसर की बनावट

1.4 8085 की बनावट

1.5 इनपुट तथा आउटपुट विधियों की एड्रेसिंग

1.6 8085 की कार्य सम्बन्धी बनावट

1.7 8085 के रजिस्टर्स

अध्याय -2

2.1 8085 के निर्देश सेट तथा उनका वर्गीकरण

2.2 एड्रेसिंग मोड

2.3 8085 की प्रोग्रामिंग

2.4 मशीनी भाषा में प्रोग्रामिंग

2.5 असेंबली भाषा में प्रोग्रामिंग

2.6 उच्च स्तरीय प्रोग्रामिंग

2.7 स्टैक

2. 8 विविध प्रोग्राम

अध्याय -3

3. 1 मेमोरी क्या है

3. 2 मेमोरी के प्रकार

3. 3 मेमोरी चिप की बनावट

3. 4 रीड तथा राइट समय चक्र

3. 5 मेमोरी तंत्र की बनावट

अध्याय -4

4.1 परिचय

4.2 समय तथा नियंत्रण इकाई

4.3 8085 मेमोरी रीड मानचित्र

4.4 8085 का मेमोरी राईट मानचित्र

4.5 रुकावट

4.5.1 इनपुट /आउटपुट रुकावट

4.5.2 8085 में रुकावट

4.5.3 RST निर्देश

4.6 अन्य 8085 रूकावट

4.7 आश्रित CALL तथा RET निर्देश

4.8 SIM

4.9 रीड रूकावट

अध्याय -5

5. 1 परिचय

5. 2 एड्रेस जगह के खाने

5. 3 मेमोरी अन्तर्मुख

5. 4 अधिक समय की गणना

5. 5 डाटा अंतरण

5. 5. 1 वर्गीकरण

5. 6 सीधे मेमोरी अधिक देरी अंतरण

5. 7 अपवत्य सीधा मेमोरी मार्ग युक्ति

5. 8 8085 तंत्र में DMA डाटा अंतरण

5. 9 क्रमानुसार डाटा अंतरण

अध्याय -6

6.1 इनपुट /आउटपुट पोर्ट

6.2 प्रोग्राम सहित इनपुट /आउटपुट पोर्ट

6.3 प्रोग्राम के घेरे में अन्तर्मुख: 8255

6.4 8255 के समय प्रचालन

6.5 प्रोग्राम रूकावट

अध्याय -7

7.1 8086 माइक्रोप्रोसेसर आर्किटेक्चर और ऑपरेशन

अध्याय -8

8.1 परिचय

8.2 मेमोरी संगठन

8.3 रजिस्टर संगठन

8.4 6800 के एड्रेस्सिंग मोड

8.5 6800 के निर्देश

अध्याय -9

9.1 परिचय

9.2 जोड़ना

9.2.1 मेमोरी से रजिस्टर को जोड़ना

9.2.2 रजिस्टर जोड़

9.2.3 तुरंत जोड़

9.2.4. क्रमानुसार जोड़

9.2.5. वृद्धि जोड़

9.2.6. केरी के साथ जोड़ना

9.3 घटाना

9.4 तुलना करना और बदलना

9.5 गुना तथा भाग

9.5.1 गुणा

9.5.2 भाग

अध्याय -10

10.1 सेंट्रल प्रोसेसिंग यूनिट

10.2 माइक्रो कंप्यूटर

10.3 कंप्यूटर सिस्टम

10.4 माइक्रोकंट्रोलर ओर एम्बेडेड प्रोसेसर

10.5 माइक्रोकंट्रोलर के प्रकार

- 10.5.1 8 बिट माइक्रोकंट्रोलर
- 10.5.2 16 बिट माइक्रोकंट्रोलर
- 10.5.3 32 बिट माइक्रोकंट्रोलर

10.6 एम्बेडेड और बाहरी मेमोरी माइक्रोकंट्रोलर

- 10.6.1 एम्बेडेड मेमोरी माइक्रोकंट्रोलर
- 10.6.2 बाहरी मेमोरी माइक्रो नियंत्रक

10.7 सीआईएससी एंड आर आईएससी आर्किटेक्चर माइक्रोकंट्रोलर

1. माइक्रोप्रोसेसर का परिचय

1.1 माइक्रो प्रोसेसर क्या है?

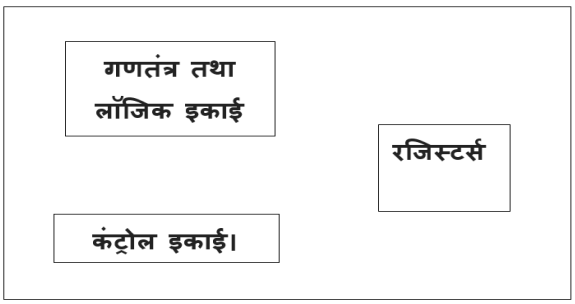
माइक्रोप्रोसेसर एक चिप होती है परंतु यह अन्य चिपों से काफी शक्तिशाली होती है। अतः यह कहा जा सकता है कि ये किसी कंप्यूटर की केन्द्रीय प्रोसेसिंग इकाई है। इसमें मेमोरी जगहों के सेट होते हैं, जिन्हें रजिस्ट्रार कहा जाता है। ये सूचनाओं को संग्रहित करते हैं और मूल आदेशों के सेट्स को ही समझते हैं। माइक्रोप्रोसेसर से बाहर सिग्नलों को उत्पन्न किया जाता है जो युक्तियों को कंट्रोल करने के काम आता है।

माइक्रोप्रोसेसर की मुख्यता तीन इकाई होती है जैसा कि चित्र (1.1) में दिखाया गया है।

-गणतंत्र तथा लॉजिक इकाई

-रजिस्टर्स

-कंट्रोल इकाई।



चित्र क्र.(1.1) माइक्रोप्रोसेसर की ईकाइयाँ

1.2 माइक्रोप्रोसेसर का इतिहास

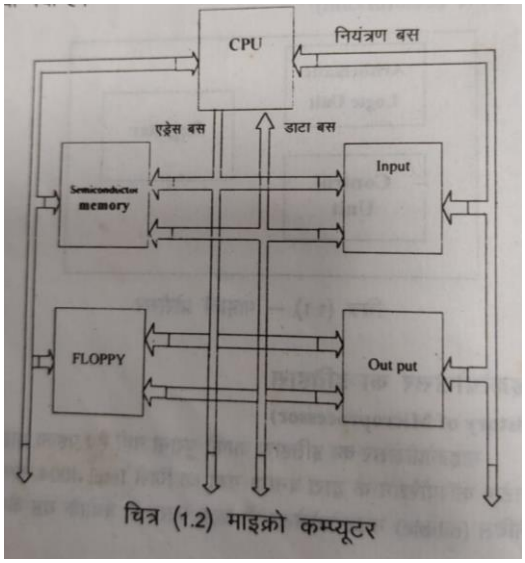
माइक्रोप्रोसेसर का इतिहास काफी पुराना नहीं है। पहला माइक्रोप्रोसेसर 1971 में इंटेल कॉर्पोरेशन के द्वारा बनाया गया था, जिससे इंटेल 4004 का नाम दिया गया। इसे निबल माइक्रोप्रोसेसर भी कहा जाता है क्योंकि ये केवल चार बिट का था और साधारण अंकगणित के लिए ही उपयोगी था। इसके बाद 8-बिट के माइक्रोप्रोसेसर बनाए गए जिसमें मुख्यता इंटेल के 8008,8085 तथा मोटोरोला का एम-6800, नेशनल सेमीकंडक्टर का एसी/एमपी ,ज़िगलोग कॉर्पोरेशन का Z80, फेयरचाइल्ड का Z80 ,हिताजी का 6809 प्रमुख हैं।

8 बिट के माइक्रोप्रोसेसर के बाद 12 और 16 बिट डेटा के माइक्रोप्रोसेसर आए। 12 बिट के माइक्रोप्रोसेसर में इंटेरसिल का आईएम 6100, तोशिबा का टी-3190 आदि प्रमुख हैं। 16 बिट के माइक्रोप्रोसेसर में फेयरचाइल्ड का 9440, इंटेल 8086, मोटोरोला 68000 आदि प्रमुख हैं।

आजकल कई कंपनियों के 32 बिट के माइक्रोप्रोसेसर बाजार में आए हैं, जिसमें आईएपी एक्स 432, इंटेल 80386 आदि प्रमुख हैं।

1.3 माइक्रो कंप्यूटर की बनावट

जैसा की हम जानते हैं की माइक्रोप्रोसेसर एक कंप्यूटर का अत्यंत उपयोगी भाग होता है। यदि हम कंप्यूटर की कल्पना करे तोह उसमे एक C.P.U , इनपुट तथा आउटपुट युक्तियाँ , मेमोरी तथा फ्लॉपी होती है जैसा चित्र में दिखाया गया है।



चित्र (1.2) माइक्रो कम्प्यूटर

चित्र (1.2) के अनुसार सभी युक्तियाँ एक दूसरे के साथ जुड़ी हुई है। यह एक तार से न जुड़कर चालक तारों के समूह से जुड़ी होती हैं, जिन्हें बस कहते हैं। अदरक किसी कंप्यूटर में बस तीन प्रकार की होती है।

-एड्रेस बस

-डाटा बस

-कंट्रोल बस।

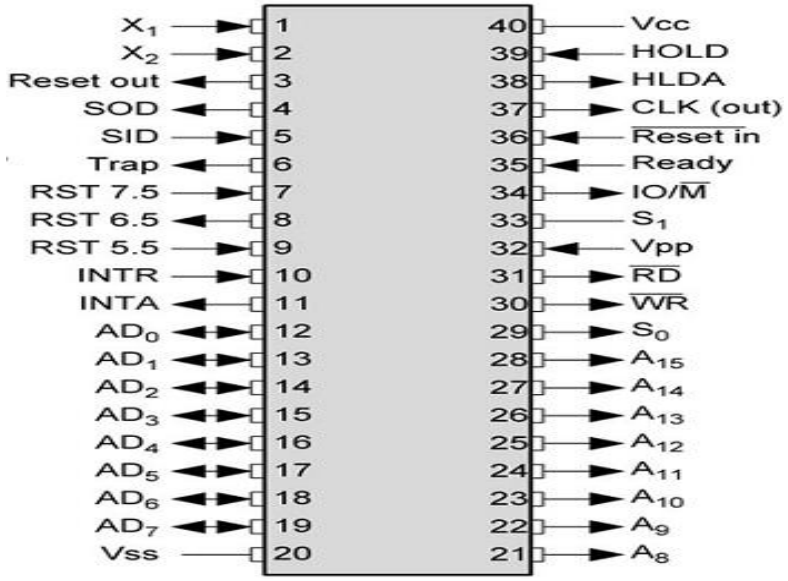
किसी भी कंप्यूटर में एड्रेस सिर्फ माइक्रो प्रोसेसर के द्वारा ही दिया जाता है। अतः ये बस एकल दिशा वाली होती है। माइक्रोप्रोसेसर को डेटा दिया भी जा सकता है और इससे डेटा लिया भी जा सकता है और ये सभी डेटा बस के द्वारा होता है। अतः यह द्विदिश वाली होती है, और उसके ,इसके अलावा ये सभी युक्तियों को नियंत्रित करती है।

1.4. 8085 की बनावट

यह एक 40 पिन कि सेरेमिक चिप होती है इसकी डाटा बस 8- बिट चौड़ी होती है अर्थात 8 बिट का डाटा माइक्रोप्रोसेसर से सामानांतर क्रम मे आ सकता है और जा भी सकता है इसमें पहले | |8 पिनो पर मेमोरी एड्रेस कि अधिकतम महत्वपूर्ण बिट जबकि सबसे कम महत्वपूर्ण बिट पर एड्रेस प्रसारित होते है ,प्रसारित होती है | डाटा ओर एड्रेस का एक साथ प्रसारित होना मल्टीप्लेक्सिंग कहलाता है |

चित्र(1.3) में 8085 की 40 पिनों को दिखाया गया है। पिन क्रमांक 1 तथा २ के मध्य एक क्रिस्टल लगाया जाता है जिसकी आवृत्ति 06 मेगाहर्ट्ज़ की होती है। पिन

क्रमांक 40 पर 5 वाल्ट का विभव लगाया जाता है। एड्रेस डाटा बस पिन क्रमांक 42 से लेकर पिन क्रमांक 49 तक होती है |डाटा बस की निम्न 8 बिट मेमोरी एड्रेस को बताते है जो प्रथम क्लॉक चक्र में दिखाई देती है। दूसरी तथा तीसरी क्लॉक चक्र में यह डाटा बस बन जाती है| एड्रेस बस पिन क्रमांक 2 से लेकर 28 तक होती है। पिन क्रमांक ३६ रिसेट इन होती है जो 8085 से जुड़ी फ्लिप फ्लोप तथा रजिस्टर्स के-डाटा को सेट करता है पिन क्रमांक ४ एंड ५ | इस समय यह लाइन कम स्थर पर होती है , जो क्रमवार आउटपुट डाटा तथा क्रमवार , एक निर्देश होते .डी.इ.तथा स.डी.ओ.स इनपुट डाटा को प्रदर्शित करते है |यहाँ ट्रैप पिन क्रमांक)6) एक अमास्केबल रूकावट होती है और ये उसी समय पहचानी जाती है जब 5.5, 6.5, और 7.5 पहचानी जाती है। पिन क्रमांक 30 अए.एल.ई. होता है। जो किसी भी वाह्य एड्रेस लैच द्वारा उपयोग की जाती है जिससे एड्रेस बिट को बन्द किया जाता है। पिन क्रमांक 29 तथा 33 बाहर की ओर क्रमान्तर रूप से भेजने वाले सिग्नल होते हैं जिनका उपयोग माइक्रोप्रोसेसर पर निर्भर करता है। S0,S1 ,.ई ओ एम्/लाइन मशीनी चक्र अवस्था कहलाती है जब S0,S1.ई ओ, एम्/० 1,कि अवस्था तो यह मेमोरी राईट कहलाता है ओर जब ये ० | ० कि अवस्था में होते है तो मेमोरी रीड कहलाता है,1,यदि ये ,1० 1,कि अवाष्ठ में है तो ये इनपुट | आउटपुट राईट कहलाता है/पिन क्रमांक 3। तथा 32 रीड तथा राईट लाइन होती है जिनका उपयोग पडनेतथा लिखने के लिए किया जाता है |पिन क्रमांक ३५ रेडी के लिए होती है| पिन ३८ तथा ३९ होल्ड तथा होल्ड पहचान पिन होती है |



चित्र(1.३) माइक्रोप्रोसेसर ८०८५ का ४० पिन डायग्राम

1.5 इनपुट तथा आउटपुट विधियों की एड्रेसिंग

जैसा कि हम जान चुके हैं कि माइक्रोप्रोसेसर न केवल इनपुट /आउटपुट विधियों से जुड़ा होता है बल्कि ये मेमोरी के साथ में जुडा होता है साथ ही मेमोरी के साथ भी जुड़ा होता है इस समय माइक्रोप्रोसेसर को यह निर्णय लेना होता है कि एड्रेस किस के लिए | | बना है

वे दो प्रकार की योजनाओं का इस्तेमाल करता है।

-मेमोरी मेप्ड इनपुट/ आउटपुट योजना/

-इनपुट,/आउटपुट मेप्ड इनपुट/ आउटपुट योजना /

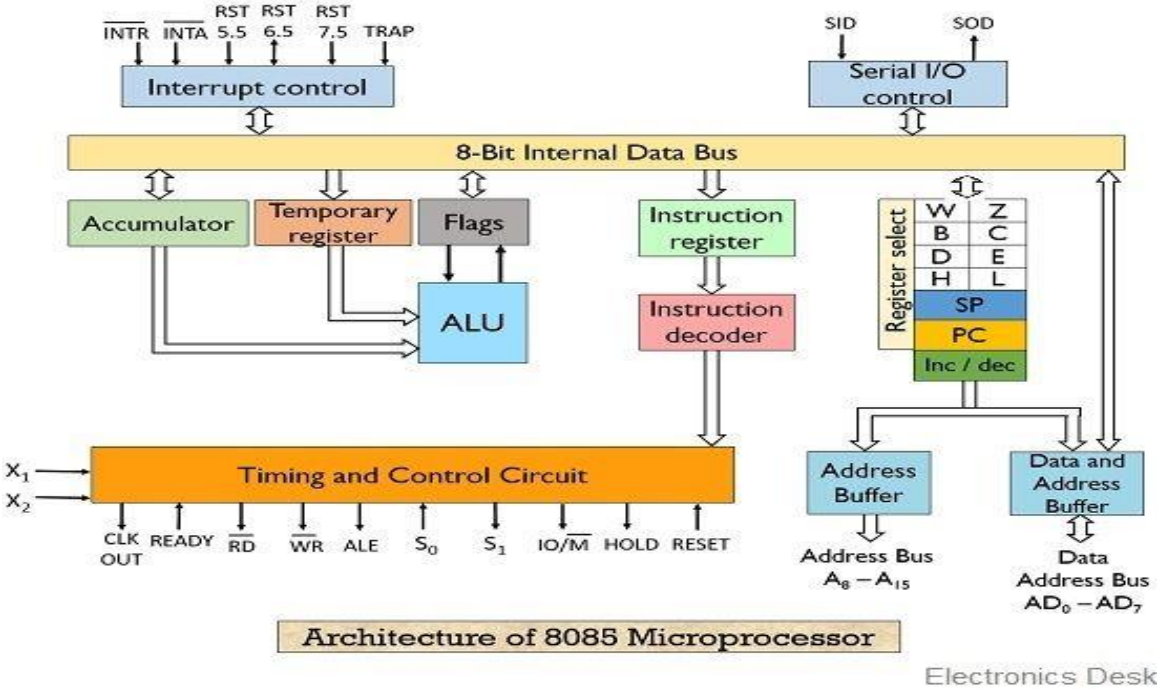
(a) मेमोरी मेप्ड इनपुट/ आउटपुट प्रणाली

अगर हम ऐसा माने कि किसी माइक्रोकम्प्यूटर में इनपुट तथा आउटपुट| विधियों को माइक्रोप्रोसेसर एक मेमोरी जगहों की तरह व्यवहार करता है तब उदाहरणके लिए यह मानें की कोई भी एड्रेस 2 माऊस के लिए है अर्थात् इनपुट के लिए है तथा एड्रेस 4, सात सिग्मेंट डिसप्ले के लिए है अर्थात् आऊटपुट के लिए है। इसका मतलब यह है कि एड्रेस 2 आने पर यह माउस के लिए है और 4 आने पर यह सात सिग्मेंट डिसप्ले के लिए है यह प्रणाली मेमोरी इनपुट ,/आउटपुट प्रणाली कहलाती है।

(b) इनपुट आउटपुट प्रणाली / आउटपुट मेप्ड इनपुट / आउटपुट प्रणाली

इस प्रणाली के द्वारा यह पता लगाया जाता है कि एड्रेस इनपुट/आउट्पुट के लिए है अथवा मेमोरी के लिए। इसके लिए माइक्रोप्रोसेसर की आईओ ऍम/ लाइन का इस्तेमाल किया जाता है। यदि आईओ ऍम/ लाइन ० होती है तो एड्रेस मेमोरी के लिए होता है और यदि यह लाइन हाई होती है तो एड्रेस इनपुट आउटपुट/ के लिए होता है |

1.6 8085 की कार्य संबंधी बनावट



चित्र(1.4) में 8085 का कार्य संबंधी बनावट दर्शायी गई है इसमें 8 बिट की आन्तरिक डाटा बस होती है जो विभिन्न प्रणालियों से एड्रेस तथा डाटा को लाती और ले जाती है। 8-बिट की संचायक होता है जो माइक्रोप्रोसेसर में एक मंच की तरह कार्य करता है इसमें किन्ीं दो डाटा में से एक का होना आवश्यक होता है। 8-बिट की अन्क्गंतीय तथा लॉजिक इकाई होती है जिसमें सभी अंकगणित तथा लॉजिकल कार्य किया जाता है इसके साथ पाँच प्लिप प्लाप के सेट- होते हैं जो अंकगणित कार्य के दौरान कैरी को रखने के काम आते है। इसके अलावा निर्देश रजिस्टर्स और डिकोडर होते हैं। 6 सामान्य उपयोगी रजिस्टर होते हैं जो इसकी मेमोरी की तरह कार्य करते है। सभी प्रणाली का समय और उनको कंट्रोल करने की इकाई होती है।

1.7 8085 के रजिस्टर

8085 में रजिस्टर्स होते हैं जो किसी प्रोग्राम को पूरा करने के लिए आवश्यक होते है। यहाँ पर एक रजिस्टर्स आठ बिट का होता है जो संचायक (ए. सी. सी) कहलाता है। इसका उपयोग कई अंकगणितीय तथा लॉजिक कार्य को पूरा करने के लिए किया जाता है। यदि मान लीजिए कि 8-बिट की दो पूर्णांक संख्याओ को घटाना है तो उनमें से एक का .(ए. सी. सी)में होना आवश्यक है, दूसरी पूर्णांक संख्या या तो किसी मेमोरी में या किसी एक सामान्य रजिस्टर में होगी।

इसके अलावा 8085 में 6 सामान्य उपयोगी रजिस्टर होते हैं। जिन्हें B,C,D,E,H तथा L कहा जाता है। इनकों अलग अलग या जोडों के रूप में इस्तेमाल किया जा सकता है। अगर अलग अलग उपयोग कर रहे है तो प्रत्येक रजिस्टर का- एक कोड होगा | यह कोड रजिस्टर 111, 000, 001, 010, 011, 100,101 होगें। जोडों के रूप में ये केवल B-C, D-E तथा H-L जोडों के लिए ही इस्तेमाल किये जा सकते है और इनके लिए कोड क्रमशः :00, 01,10 होगें।

2. 8085 के निर्देश सेट तथा वर्गीकरण

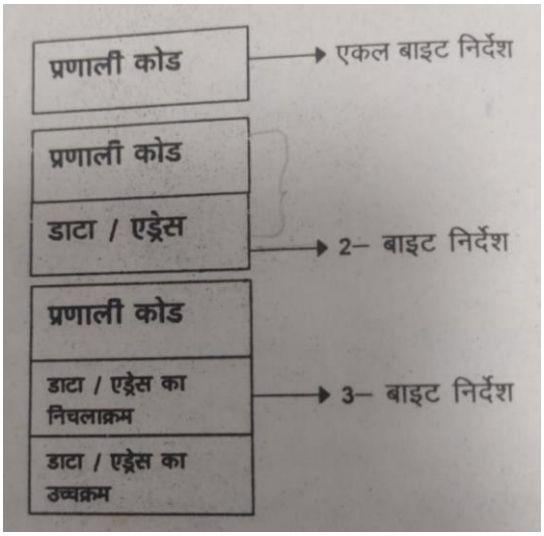
2.1 8085 के निर्देश सेट तथा उनका वर्गीकरण

निर्देश सेट में एक प्रणाली कोड तथा एक ओपरेंड होता है। प्रणाली कोड यह बताता कौन सी प्रणाली पर कार्य होना है तथा ओपरेंड बताता है कि किस डाटा तथा एड्रेस पर यह कार्य होना है। जैसा चित्र)2.) में दिखाया गया है कि निर्देश एक दो या तीन बाइट लम्बे हो सकते है। अगर एक बाइट निर्देश है तो इसमें सिर्फ प्रणाली कोड ही होगा। यदि 2-बाइट निर्देश है तो प्रणाली कोड के साथ ओपरेंड भीहोगा जिसमें या तो डाटा या एड्रेस होगा। 3-बाइट निर्देश सेट में प्रणाली कोड के साथ 2-बाइट के ओपरेंड होंगे। पहली बाइट में प्रणाली कोड होगा। दूसरी बाइट में एड्रेस तथा डाटा की बाइट का निचला क्रम तथा तीसरी बाइट में एड्रेस डाटा बाइट का उच्चक्रम आता हैं।

-एकल बाइट निर्देश

-2- बाइट निर्देश

-3- बाइट निर्देश



प्रोग्राम को और सरलीकृत करने के लिये 8085 के निर्देशों का पाच समहों में वर्गीकृत किया जाता है |

-डाटा अन्तरण समूह

-अंकगणित समूह

-लॉजिकल समूह

-शाखा समूह

-स्टेक, इनपुट/आउटपुट तथा

-मशीन नियंत्रण समूह

2.2 एड्रेसिंग मोड

एक निदेश, एक प्रणाली को निश्चित डाटा पर कार्य करने के लिये दिया जाता है। डाटा के एड्रैस को प्रचालित करनेके लिये विभिन्न यंत्र कला का प्रयोग किया जाता है जिन्हें- एड्रेसिंग मोड कहते हैं।8085 में निम्न चार को एड्रेसिंग मोड का इस्तेमाल किया जाता है।

-सीधी एड्रेसिंग

-रजिस्टर एड्रेसिंग

-रजिस्टर अप्रत्यक्ष एड्रेसिंग

-तुरंत एड्रेसिंग

-अनुमित एड्रेसिंग

सीधी एड्रेसिंग

इस प्रकार कि एड्रेसिंग में ओपरेंड प्रणाली कोड में ही निहित होता है .| उदाहरण के लिये यदि किसी मेमोरी जगह से किसी सूची को दूसरी मेमोरी जगह में अन्तरण करना है तो सीधी एड्रेसिंग का प्रयाग किया जाता है। यह एड्रेसिंग दो या तीन बाइट लम्बी होती है जो एक ही निर्देश में निहित होती है जैसे LDA, STA

रजिस्टर एड्रेसिंग

इस प्रंकार की एड्रेसिंग में किसी भी प्रणाली का ओपरेंड
किसी सामान्य इस्तेमाल के रजिस्टर में हो तो सिर्फ उस रजिस्टर्स को ही बताना होता है यह निर्देश एक बाइट के होते हैं जैसे MOV,ADD

रजिस्टर अप्रत्यक्ष एड्रेसिंग

रजिस्टर अप्रत्यक्ष एड्रेसिंग में किसी इंगित किये हुये रजिस्टर की सूची उस का एड्रेस मानी जाती है। इस मोड में एक रजिस्टर को इंगित करने के स्थान पर एक रजिस्टर युग्म इंगित किया जाता हैं।उदाहरण के लिये इसमें मेमोरी जगह वह अगित होगी जो H-L यग्म द्वारा इंगित की जायेगी अर्थात् M का मतलब H-L युग्म द्वारा इंगित की गई मेमोरी जगह।

तुरंत एड्रेसिंग

इस तरह के संग मोड में ओपरेंड/निर्देश में ही संहित रहता है। इसमें सिर्फ प्रणाली कोड का ही इस्तेमाल किया जाता है, उदाहरण के लिये LXI,LDA:

अनुमित एड्रेसिंग

इसमें निर्देश सिर्फ एक ओपरेंड द्वारा प्रणालित होता है । इसमें डाटा एड्रेस में ही होता है। उदाहरण के लिये CMA

2.3 8085 की प्रोग्रामिंग

प्रोग्राम निर्देशों का एक क्रम होता है और प्रोग्रामिंग एक विधि होती है । प्रोग्राम बनाने में एक भाषा का इस्तेमाल किया जाता है माइक्रोप्रोसेसर सिर्फ 1 और 0 की भाषा को समक्षता है। प्रोग्रामर जिस भाषा में प्रोग्राम लिखता है वह एस्सेम्ब्ली भाषा कहलाती है माइक्रोप्रोसेसर इसको ० 1,में बदलता है | इसे मशीनी भाषा कहते है|

एक प्रोग्राम को लिखने के लिए इत्तके अल्गोरिथिम्स लिखी जाती है जिसमे प्रोग्राम के यथार्थ कथन होते है |

८०८५ में अंग्रेजी के प्रणाली कोडों तथा अंको के लिये हेक्सा डेसीमल नंबर का इस्तेमाल किया जाता है इसके पश्चात यह निर्देश कोड होकर ० तथा १ में बदलकर माइक्रोप्रोसेसर द्वारा पढ़े जाते है।

अगर प्रोग्राम मशीनी भाषा में लिखे जाते है तो ये माइक्रोप्रोसेसर मेमोरी जगहों में भेजे जाते हैं। जब यह पूरी तरह भर जाते हैं तो माइक्रोप्रोसेसर पहले निर्देश से इन्हें हल कर शुरू करता है और हमें सिग्नल प्राप्त होने शुरू हो जाते हैं।

2.4 मशीनी भाषा में प्रोग्रामिंग

(मशीन भाषा में प्रोग्रामिंग)

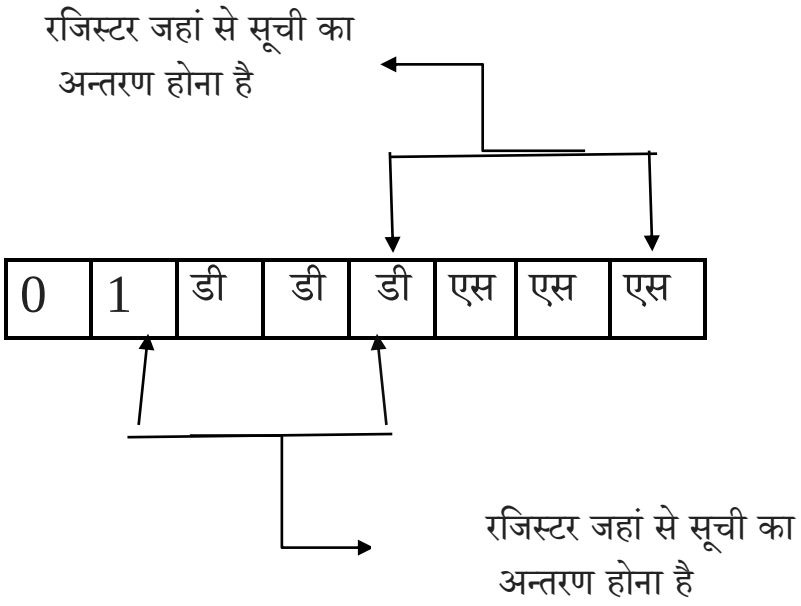
मशीन भाषा प्रोग्रामिंग में बाइनरी (0 और 1) तथा हेक्साडेसीमल कोड (Hexadecimal Code) का इस्तेमाल किया जाता है। माइक्रोप्रोसेसर का अपना एक निर्देश सेट होता है जो उसके शिल्पकारों (Manufacturer) पर निर्भर करता है। किन्हीं दो माइक्रोप्रोसेसर के निर्देश सेट एक जैसे नहीं हो सकते हैं।

डाटा अन्तर समूह (Data transfer group) में

(MOV r1, r2 (MOVE REGISTER)

यहाँ इस मेनेमोनिक्स का अर्थ है कि रजिस्टर 12 की सूची को r1 में

ले जाना। इसको यदि मशीनी भाषा में लिखना है तो हमें 8 बिट के निम्न फोरमेट का इस्तेमाल करना होगा। (चित्र 2.2)



चित्र (2.2) रजिस्टर से रंजिस्टर में अन्तरण का खाका

उदाहरण :-

MOVA,C: C रजिस्टर की सूची (Contents) को जिस्टर A में अन्तरण इसको मशीन कोड में निम्न प्रकार लिखा जायेगा -

01

111

0	1	डी	डी	डी	एस	एस	एस

रजिस्टर A का कोड

001

रजिस्टर C का कोड

डाटा

इसी तरह यदि मेमोरी से किसी सूची को रजिस्टर में अन्तरण करन है तो मशीन भाषा में निम्न फोरमेट निम्न होगा :-

MOV R, M; मेमोरी से सूची **(Contents)** का रजिस्टर में अन्तरण

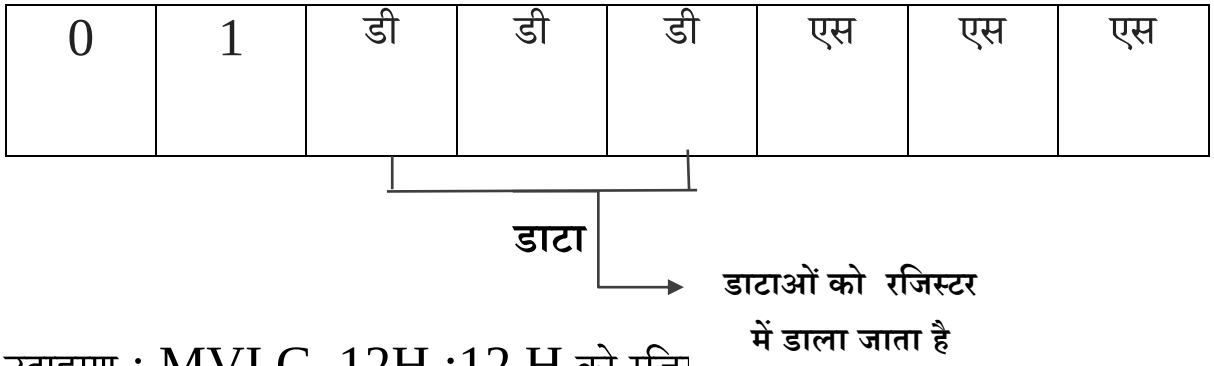
रजिस्टर जहाँ मेमोरी में सूची का अंतरण होता है

चित्र (2.3)मेमोरी से रजिस्टर में अंतरण का फॉर्मेट

उदाहरण : MOV D ,M :H -L युग्म से निर्देशित मेमोरी में सूची का अंतरण रजिस्टर D में

यदि सीधे डाटा को किसी रजिस्टर मे लाना हो तोह मशीनी भाषा में निम्न फॉर्मेट का इस्तेमाल करना पड़ता है जैसा चित्र 2.4 मे दिखाया गया है ।

0	1	डी	डी	डी	एस	एस	एस



डाटा

डाटाओं को रजिस्टर में डाला जाता है

उदाहरण : MVI C ,12H ;12 H को रजिस

00   0 0 1 - 1 1 0   0 0 0 1 D D 1 0

रजिस्टर C का कोड डाटा   जिसमे डाटा डाला जाना है।

इस तरह मशीनी भाषा प्रोग्रामिंग में 0 तथा 1 को इस्तेमाल मे लेकर प्रोग्राम बनाये जाते है ।ये बनाने में , अत्यंत कठिन होते हैं तथा काफी समय लगता हैं साथ ही गलतियां होने की संभावना भी अधिक होती हैं ।माइक्रोप्रोसेसर बनाने वाली हर कंपनी के कोड अलग होते हैं और यदि मशीन भाषा मे प्रोग्राम बनाना है तो कोड सूची की आवश्यकता पड़ती है ।

2 .5 असेंबली भाषा मे प्रोग्रामिंग

असेंबली भाषा मे कोई प्रोग्राम बनाया जाता है तोह उसके चार क्षेत्र होते है जो निम्न है :

(१) लेबिल

(२) निर्देश, स्मृति -सहायक या प्रणाली कोड

(३) ऑपरेंड क्षेत्र

(४) व्याख्या क्षेत्र

इन चार क्षेत्रों को नीं प्रकार से लिखा जाता है:

लेबिल	स्मृति सहायक	ऑपरेंड	व्याख्या
स्टार्ट	MOV ADI JMP	A,9H 20H स्टार्ट	9_{16} को रजिस्टर A में डालना 20_{16} को ACC की सूची के साथ जोड़ना प्रोग्राम के शुरुआत में जाना

इस तरह असेंबली भाषा में प्रोग्रामिंग , मशीनी भाषा में ज्यादा सरल होती है ।प्रत्येक स्मृति सहायक यह बताता है की प्रोग्राम किस तरह प्रचलित होगा ।

2.6 उच्च स्तरीय प्रोग्रामिंग:

असेंबली भाषा में प्रोग्रामिंग करके निश्चित रूप से प्रोग्रामर की क्षमता बढ़ती है क्योंकि मशीनी भाषा में कोड को लिखना तथा उन्हें याद करना एक अत्यंत कठिन कार्य है ।

उच्च प्रोग्रामिंग मे प्रोग्रामर को प्रत्येक प्रणाली के लिए एक प्रणाली कोड दिया जाता है और मॉइक्रोप्रोसेस्सोर उन प्रणाली कोडो को क्रमबद्ध तरीके से लेता है ।बाद मे उच्च स्तरीय प्रणाली कोड का बाइनरी कोड मे परिवर्तित किया जाता है ।

FORTRAN, COBOL, BASIC, C++, PASCAL इसके कुछ उदाहरण है।

2.7 स्टेक

यह मइक्रोप्रोसेससर में 16 बिट का एक रजिस्टर होता है। यह एक मेमोरी की तरह कार्य करता है जिसकी प्रकृति अंत का अंदर पहला बाहर की होती है। अर्थात जो भाग बाद में जायेगा वह पहले बाहर जायेगा। यह इसलिए भी होता है क्युकी स्टेक READ तथा WRITE प्रणाली को एक सिरे से करता है।

स्टेक दो प्रकार के होते है

1. हार्डवेयर स्टेक

2. सॉफ्टवेयर स्टैक

हार्डवेयर स्टैक का उपयोग करने के लिए अत्यधिक तीव्र गति के रेगिस्टर्स की आवश्यकता होती है जिससे तीव्र जवाब प्राप्त होता है।

सॉफ्टवेयर स्टैक का उपयोग करने के लिए कई जगहों का इस्तेमाल किया जाता है।

सटक का उपयोग सामान्य रूप से सबरूटीन के लिए किया जाता है यहाँ ये जान लेना जरुरी है की सबरूटीन एक प्रोग्राम है जो मुख्या प्रोग्राम द्वारा समय समय पर उपयोग किया जाता है।

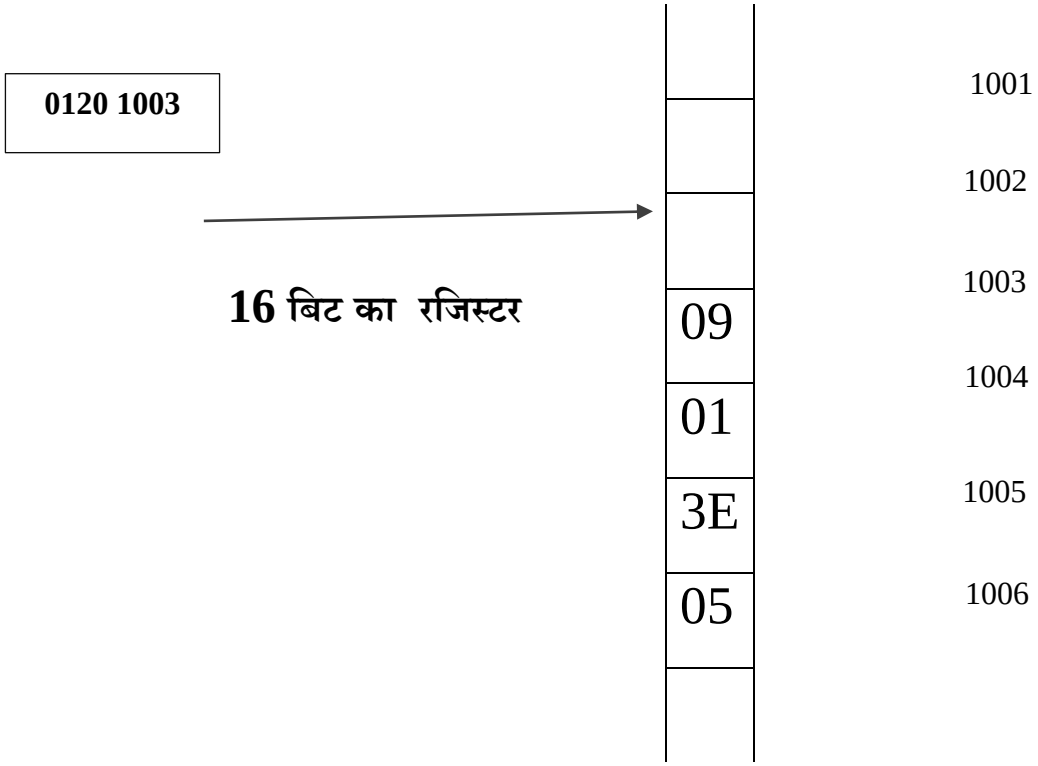
एक स्टैक को प्रचालित करने के लिए दो निम्न निर्देशों का उपयोग किया जाता है।

1. पुश निर्देश

2. पॉप निर्देश

पुश निर्देश :

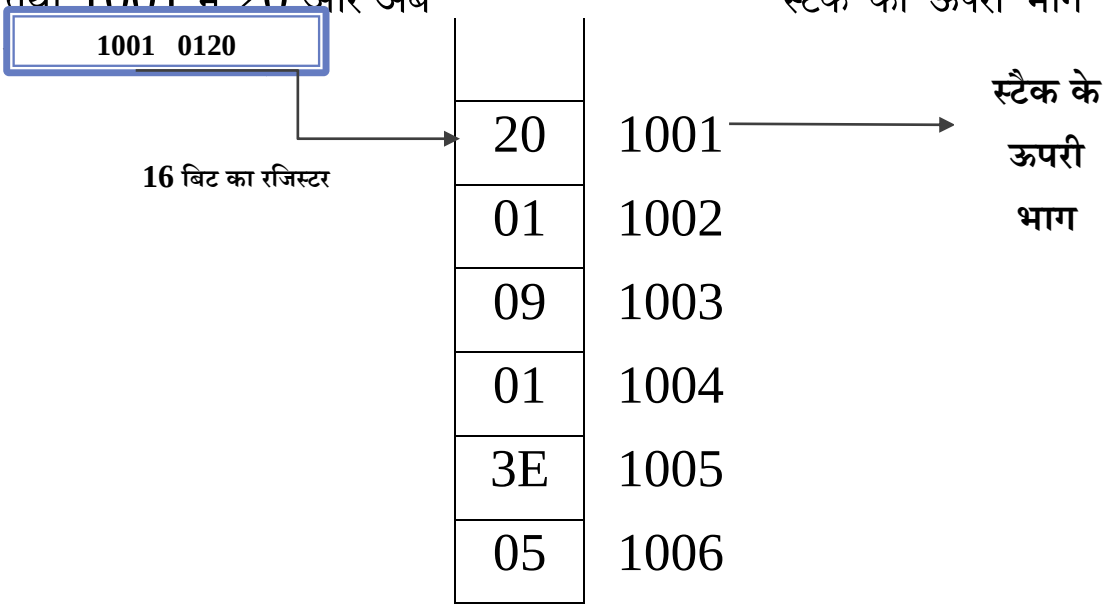
पुश निर्देश ,एक लिखने की क्रिया होती है जो स्टैक के ऊपरी तथा निचले जगहों पर लिखे जाते है। इसे एक उदहारण की सरलता से समझा जा सकता है। पुश निर्देश देने से पहले इस स्थिति पर विचार है।



जैसा चित्र (2.5) में दिखाया गया है की एक 16 बिट में रखे डाटा को स्टैक में पुश निर्देश की सहायता से डालना है। इस स्थिति में स्टैक का ऊपरी भाग 1003 को निर्देशित करेगा। 16 बिट रजिस्टर किन्ही 2 सामान्य उपयोगी रजिस्टर के युग्म भी हो सकते है।

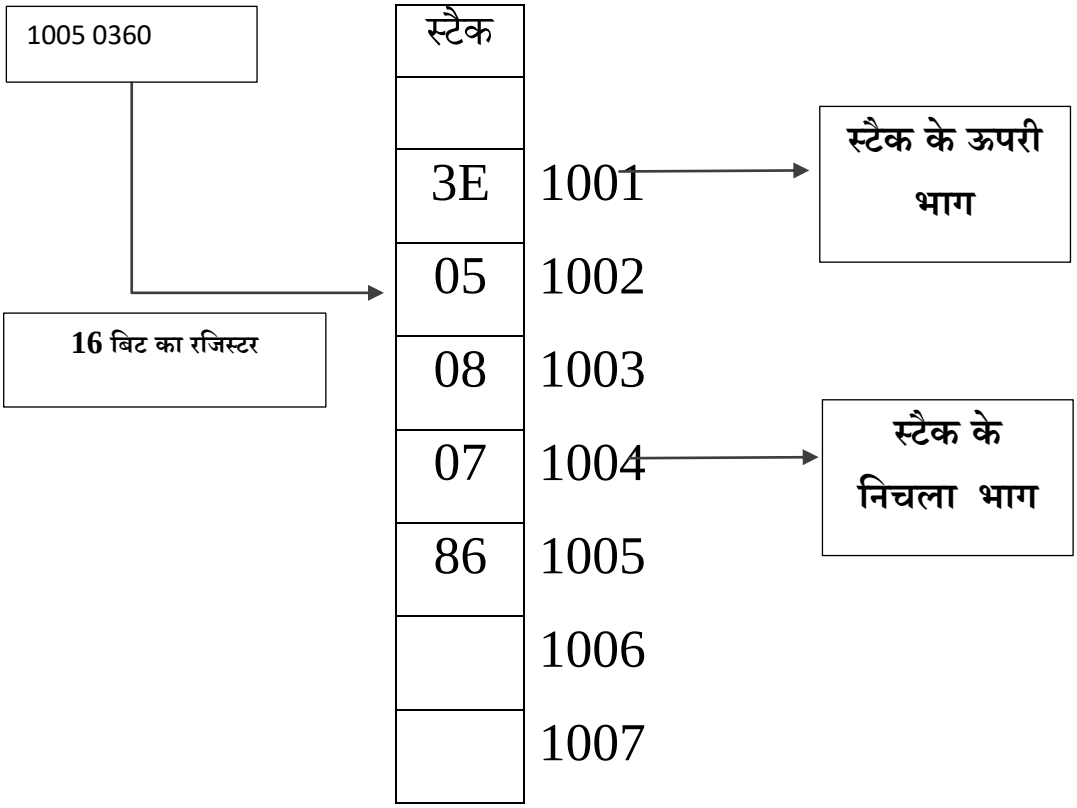
अब पुश निर्देश देने के पश्चात की स्थिति पर विचार करते है।

जैसा की (2.6) दर्शाया गया है की इस पुश निर्देश देने के पश्चात 16 बिट के रजिस्टर सूची में क्रमशः स्टैक के ऊपरी भाग की जगहों में लिख जाती है जैसे 1002 में 01 तथा 1001 में 20 और अब स्टैक का ऊपरी भाग 1001

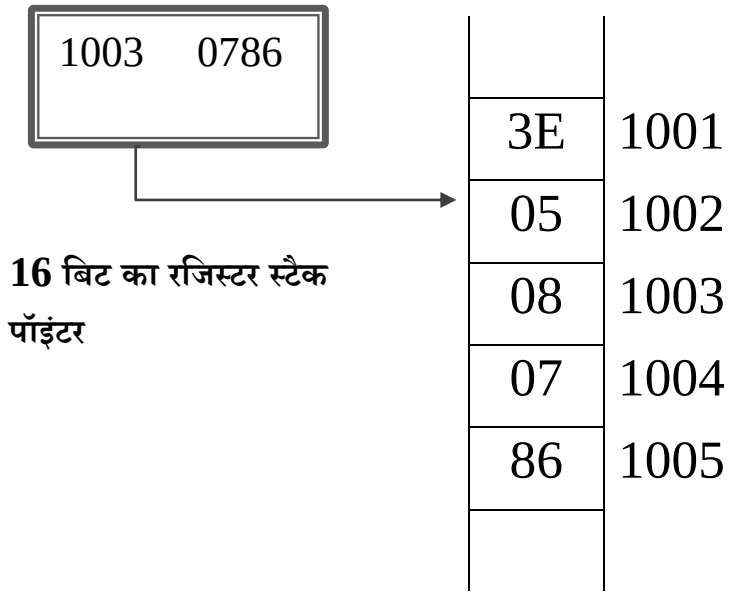


पॉप निर्देश :

पॉप निर्देश स्टैक के ऊपरी तथा निचले सिरे से पढ़ने के काम आता है। इसे निम्न उदाहरण की सहायता से समझते है। पहले पॉप निर्देश देने के पहले कि स्थिति को समझते है।



POP निर्देश देने के पश्चात स्टैक के अंतिम मेमोरी जगहों से सूची 16 बिट के रजिस्टर में चला जाता है तथा स्टैक पॉइंटर की निचली स्थिति को 1003 हो जाती है जो निर्देश के पूर्व 1005 थी। जिसे चित्र 2. 8 में दिखाया गया है।



2.8 विविध प्रोग्राम

1. मेमोरी स्थान 20E2 की सामग्री को पूरक करने के लिए

MEM एड्रेस	डाटा	स्मृति सहायक	व्याख्या
7000	21	LXIH,20E2	मेमोरी लोकेशन 20E2 के साथ H-L पेयर लोड करें
7001	E2		
7002	20		
7003	7E	MOV A, M	स्थान की सामग्री को एसीसी में ले जाएं
7004	2F	CMA	एसीसी की सामग्री का पूरक
7005	77	MOV M, A	एसीसी की सामग्री को स्मृति में ले जाएं
7006	CF	RST-1	कमांड मोड पर लौटें

2. 16 बिट सामग्री के साथ लोड रजिस्टर जोड़ी

MEM एड्रेस	डाटा	स्मृति सहायक	व्याख्या
7000	21	LXI H, 1532	डेटा 1532 के साथ लोड रजिस्टर पेयर
7001	32		
7002	15		
7003	CF	RST	कमांड मोड पर लौटें

3. एचएल जोड़ी द्वारा बताई गई मेमोरी लोकेशन में सामग्री 20C2 लोड करने के लिए

MEM एड्रेस	डाटा	स्मृति सहायक	व्याख्या
7000	21	LXI, H 20 C2	स्मृति स्थान 20C2 के साथ लोड एच-एल जोड़ी
7001	C2		
7002	20		
7003	36	MVI M	स्मृति के साथ सामग्री 15 लोड करें
7004	15		
7005	CF	RST	कमांड मोड पर लौटें

4. 03H सामग्री को ACC में लोड करें और 52H डेटा बाइट के साथ C को पंजीकृत करें और संचायक के साथ रजिस्टर C के बिट को तार्किक रूप से ऑर्डर करें।

MEM एड्रेस	डाटा	स्मृति सहायक	व्याख्या
7000	3E	MVI A,03 H	A को 03H के साथ लोड करें
7001	03		
7002	0E	MVI C,52H	C को 52H के साथ लोड करें
7003	52		
7004	B1	OR AC	तार्किक रूप से या सी ए के साथ
7005	47	MOV B, A	ए को ओ में स्थानांतरित करें
7006	CF	RST	कमांड मोड पर लौटें

5. संचायक और पंजीयक डी के पास क्रमश: 54एच और 82एच हैं तार्किक रूप से एसीसी की सामग्री के साथ रजिस्टर डी की सामग्री को संपादित किया गया है।

MEM एड्रेस	डाटा	स्मृति सहायक	व्याख्या
7000	3E	MVI A,54 H	A को 54 से लोड करें
7001	54		
7002	16	MVI D,82	D को 82 के साथ लोड करें
7003	82		
7004	A2	AN A D	तार्किक रूप से और ए के साथ डी
7005	47	MOV B, A	ए को बी में ले जाएं
7006	CF	RST 1	कमांड मोड पर लौटें

6. दो 8-बिट संख्या घटाना और परिणाम को किसी भी सामान्य उद्देश्य रजिस्टर में संग्रहित करना।

MEM एड्रेस	डाटा	स्मृति सहायक	व्याख्या
7000	3E23	MVI A, 23H	एसीसी को 23 से लोड करें।
7002	OE	MVI C, 11H	रजिस्टर सी में डेटा 11एच लोड करें।
7003	11		
7004	91	SUB C	एसीसी से रजिस्टर सी की सामग्री घटाएं।
7005	47	MOV D, A	परिणाम को रजिस्टर डी में स्थानांतरित करें।
7006	CF	RST -1	कमांड मोड पर लौटें।

7. Exclusive OR रजिस्टर की सामग्री के लिए

MEM एड्रेस	डाटा	स्मृति सहायक	व्याख्या
7000	3E	MVI A, 11H	एसीसी को डेटा 11 के साथ लोड करें।
7001	11		
7002	06	MVI B, 12	डेटा 12 के साथ लोड रजिस्टर बी।
7003	12		
7004	A8	XRA, B	Exclusive OR सामग्री और रजिस्टर ए और बी।
7005	CF	RST -1	कमांड मोड पर लौटें।

8. रजिस्टरों और मेमोरी स्थानों की सामग्री के बीच AND परिणाम दिखाने के लिए एक प्रोग्राम लिखें।

MEM एड्रेस	डाटा	स्मृति सहायक	व्याख्या
7000	21	LXI H, 20 C0	मेमोरी एड्रेस 20C0 के साथ H-L पेयर रजिस्टर लोड करें।
7001	CO		
7002	20		
7003	36	MVI M, 15	डेटा 15 को मेमोरी में लोड करें।
7004	15		
7005	3E	MVI A, 02	मेमोरी में डेटा 02 लोड करें।
7006	02		
7007	A6	ANA M	AND एसीसी की सामग्री स्मृति की सामग्री के लिए और परिणाम एसीसी में संग्रहीत करें।
7008	CF	RST - 1	कमांड मोड पर लौटें।

9. दो 8-बिट हेक्साडेसिमल संख्याओं 11H और 04H को गुणा करने के लिए।

MEM एड्रेस	डाटा	स्मृति सहायक	व्याख्या
7000	3E	MVI A, 11H	लोड डेटा 11H को गुणा किया जाना है।
7001	11		
7002	47	MVI B, 12	रजिस्टर ए की सामग्री को रजिस्टर बी में ले जाएं।
7003	3E	MVI A,04	रजिस्टर ए की सामग्री को रजिस्टर सी में ले जाएं।
7004	04		
7005	4F	MOV C, A	रजिस्टर ए की सामग्री को सी में ले जाएं।
7006	3E	MVI A,004	एसीसी साफ़ करें।
7007	00		
7008	81	Loop, ADD C	एसीसी ए में रजिस्टर सी की सामग्री जोड़ें।
7009	05	DCR B	B को 1 से कम करें।
700A	C2	JNZ Loop	यदि शून्य है
700B	08		
700C	70		
700D	CF	RST	कमांड मोड पर लौटें।

10. दो मेमोरी स्थान घटाना और परिणाम को रजिस्टर सी में स्टोर करना।

MEM एड्रेस	डाटा	स्मृति सहायक	व्याख्या

7000	01	LXI B, 20C5H	बीसी रजिस्टर जोड़ी में ऐड मेमोरी लोकेशन लोड करें।
7001	C5		
7002	20		
7003	0A	LDA X, 13	स्मृति स्थान की सामग्री लोड करें जहां बीसी में पता विशिष्ट है।
7004	21	LXI H, 2066	एच-एल जोड़ी में अगली मेमोरी लोकेशन का पता लोड करें।
7007	46	MOV A, M	20C6 की सामग्री को रजिस्टर बी में ले जाएं।
7008	98	SUB B	ए की सामग्री घटाना
7009	4F	MOV C, A	परिणाम को सी में स्थानांतरित करें।
700A	CF	RST- 1	कमांड मोड पर लौटें।

11. दो 16-बिट संख्याएँ जोड़ने के लिए और परिणाम को H-L जोड़ी रजिस्टरों में संग्रहीत करने के लिए।

MEM एड्रेस	**डाटा**	**स्मृति सहायक**	**व्याख्या**
7000	21 11 21	LXI H, 1121	16 बिट डेटा 1121 के साथ एच-एल जोड़ी लोड करें।
7003	01 02 12	LXI B, 1202	डेटा 1202 के साथ लोड रजिस्टर जोड़ी बी-सी।
7006	09	Add B	रजिस्टर जोड़ी की सामग्री जोड़ें जोड़ी एचएल रजिस्टर करने के लिए जोड़ी बी-सी। और एच-एल में परिणाम स्टोर करें।
7007	CF	RST - 1	कमांड मोड पर लौटें।

12. दो 8-बिट नंबर जोड़ने के लिए और परिणाम को किसी भी रजिस्टर में स्टोर करने के लिए।

MEM एड्रेस	डाटा	स्मृति सहायक	व्याख्या
7000	3E	MVI A	A को 11H के साथ लोड करें।
7001	11		
7002	06	MVI B	12H के साथ B लोड करें।
7003	12		
7004	80	Add B	B को A से जोड़ें।
7005	57	MOV D, A	A को D में ले जाएं।
7006	CF	RST -1	सभी रजिस्टरों को सहेजते हुए कमांड मोड पर जाएं।

13. स्मृति स्थान 20C2 और 20C3 की सामग्री को जोड़ने के लिए और परिणाम को स्मृति स्थान 20C4 में रखें।

MEM एड्रेस	डाटा	स्मृति सहायक	व्याख्या
7007	21 C220	LXI H, 20C2	एच-एल जोड़ी में स्मृति स्थान 20C2 का पता लोड करें।
7003	7E	MOV A, M	स्मृति स्थान 20C2 की सामग्री को एसीसी में ले जाएं।
7004	23	INX H	H-L पेयर को 1 से बढ़ाएँ।
7005	86	ADD M	स्मृति स्थान 20C3 की सामग्री को एसीसी में जोड़ें
7006	23	INX H	H-L पेयर को 1 से बढ़ाएँ

7007	77	MOV H, A	A की सामग्री को मुख्य स्थान 20C4 पर ले जाएँ
7008	CF	RST-1	कमांड मोड पर लौटें।

14. मेमोरी डेटा के बिट-4 के परीक्षण के लिए एक प्रोग्राम लिखें।

MEM एड्रेस	**डाटा**	**स्मृति सहायक**	**व्याख्या**
7000	21	LXI H,20DO	मेमोरी लोकेशन 20DO के साथ H-L पेयर लोड करें
7001	D0		
7002	20		
7003	36	MVI M, 15	डेटा 15 के साथ मेमोरी लोड करें
7004	15		
7005	7E	MOV A, M	स्मृति को एसीसी में स्थानांतरित करें
7006	E6	ANI 08	और डेटा के साथ मेमोरी 08
7007	08		
7008	CA	JZ 700B	अगर A=00[zflay=1] तो बिट 4=0 ELSE बिट 4=1
7009	0B		
700A	70		
700B	CF	RST	कमांड मोड पर लौटें

15. बड़ा नंबर खोजने के लिए दो 8 बिट संख्या के बीच

MEM एड्रेस	**डाटा**	**स्मृति सहायक**	**व्याख्या**

7000	06	LXI H, 20 DO	मेमोरी लोकेशन 20DO के साथ H-L पेयर लोड करें
7001	10		
7002	3E		
7003	02		
7004	B8	CMP	डेटा 15 के साथ मेमोरी लोड करें
7005	D2	JNC 7009	स्मृति को एसीसी में स्थानांतरित करें
7006	09		और डेटा के साथ मेमोरी 08
7007	70		
7008	78	MOV A, B	अगर A=00[zflay=1] तो बिट 4=0 ELSE बिट 4=1
7009	CF	RST - 1	कमांड मोड पर लौटें

16. रजिस्टर जोड़ी एच-एल और डी-ई की सामग्री का आदान-प्रदान करने के लिए एक कार्यक्रम लिखें।

MEM एड्रेस	**डाटा**	**स्मृति सहायक**	**व्याख्या**
7000	21	LXI H 20 C0	एच-एल जोड़ी को 20C0 के साथ लोड करें
7001	C0		
7002	20		
7003	11	LXI D 20 C1	20C1 के साथ D-E जोड़ी लोड करें
7004	C1		
7005	20		
7006	EB	XCHG	एच-एल जोड़ी और डी-ई जोड़ी की सामग्री का आदान-प्रदान करें
7007	CF	RST	कमांड मोड पर लौटें

17. स्थान 20C0 में संग्रहीत 8-बिट शब्द पर समता जाँच करने के लिए एक प्रोग्राम लिखें। यदि समता एक ही स्थान पर विषम स्टोर 000H है। हालाँकि यदि समानता 20C0H स्थान पर स्टोर 0EEH की जाँच भी है।

MEM एड्रेस	डाटा	स्मृति सहायक	व्याख्या
7000	3A, C020	LDA 20 C0H	A में 8 बिट डेटा लोड करें
7003	C600	ADI 00H	फ्लैग को प्रभावित करने के लिए A में 00H जोड़ें
7005	E20E70	JPO ODD	प्रति ODD जोड़ी की जाँच करें
7008	3E	MVI A, EE	अगर समता OEEH भी है
700A	32C020	STA 20C0H	A से मेमोरी लोकेशन 20CO में 8 बिट स्टोर करें
700B	CF	RST – 1	कमांड मोड पर लौटें
700E	3E	MVI A, ODDH	यदि स्मृति स्थान 20C0H में समता विषम है
7010	32 C020	STA 20C0	A से 20C0H के 8 बिट को स्टोर करें
7013	CF	RST -1	कमांड मोड पर लौटें

18. मेमोरी लोकेशन 20C0H में स्टोर नंबर पॉजिटिव है या नेगेटिव, यह चेक करने के लिए एक प्रोग्राम लिखें।

MEM एड्रेस	डाटा	स्मृति सहायक	व्याख्या
7000	21, C020	LXI H, 20C0H	एचएल जोड़ी को 20C0H के साथ लोड करें
7003	7E	MOV A, M	मेमोरी सामग्री को एसीसी में स्टोर करें

7004	17	RAL	कैरी के माध्यम से ए की सामग्री को घुमाएं
7005	DA	0BJ0 JC Minus	ले जाने के लिए जाँच करें
7008	3E	MVI A, ODDH	अगर डेटा ODDH के साथ कोई कैरी लोड नहीं है
700A	CF	RST – 1	कमांड मोड पर लौटें
700B	3EEE	Minus; MVI 0EEH	यदि ए, ओईईएम के साथ कैरी लोड ए है
700D	CF	RST – 1	कमांड मोड पर लौटें

19. यह जाँचने के लिए एक प्रोग्राम लिखें कि क्या स्मृति स्थान में संग्रहीत 8-बिट डेटा 20C0H एक सम संख्या है या विषम संख्या।

MEM एड्रेस	**डाटा**	**स्मृति सहायक**	**व्याख्या**
7000	3A	LDA 20C0H	एसीसी में 8 बिट डेटा लोड करें
7001	C0		
7002	20		
7003	IF	RAR	कैरी के माध्यम से ए की सामग्री को घुमाएं
7004	DA OA70	JC ODD	ले जाने के लिए जाँच करें
7007	3E	MVI A OEEH	यदि कैरी को एसीसी में लोड ओईई रीसेट किया जाता है।
7008	EE		
7009	CF	RST – 1	कमांड मोड पर लौटें

3. मेमोरी और उसके प्रकार

मेमोरी क्या है?

जैसा कि हम सभी जानते हैं कि फ्लिप फ्लोप (Flip-Flop) के द्वारा लॉजिकल O'S तथा 1'S को स्टोर किया जाता है। जब तक कि फ्लिप फ्लोप (Flip. Flop) की अवस्था में परिवर्तन न किया जाये उनमें स्टोर O'S तथा 1'S को परिवर्तित नहीं कियाजा सकता है।

यदि हमें कई 1 तथा कई 0 को स्टोर करना हो तो हमें बहुत सारे फ्लिप फ्लोप को एक IC (Integrated Circuit) में रखना होगा और इस तरह से जो मेमोरी चिप तैयार होगी उसे अर्द्धचालक मेमोरी कहेगें। अर्द्धचालक मेमोरी एक पूरे शब्द (Word) के साथ व्यवस्थित की जाती है। जैसा कि हम जानते हैं कि डिजिटल इलेक्ट्रानिक में 8 बिट, 1 बाइट या एक शब्द को बनाते हैं। अगर हम एक ऐसी मेमोरी की कल्पना करे जिसमे 000 से FFF तक हैक्सा डेसीमल निर्देश हो तो यह मेमोरी 4096 शब्दो को व्यवस्थित करेगी। अर्थात् यह एक 4096 8 बिट मेमोरी बनायेगी।

माइक्रोप्रोसेसर की मेमोरी में एक शब्द को रखने के लिए कई रजिस्टर्स होते है लेकिन रजिस्टर्स के लिए स्थान की जरूरत होती है इसीलिए माइक्रोप्रोसेसर में कुछ ही रजिस्टर्स रखे जाते हैं। उदाहरण के लिये 8085 में 6 सामान्य उपयोगी रजिस्टर्स होते है।

सामान्य रूप से माइक्राप्रोसेसर में कई प्रोग्रामो को एक साथ रखना पड़ता है इसलिए जो प्रारम्भिक मेमोरी होती है उसकी गति के बारे में हमें सोचना पड़ता है।

कभी-कभी इन मेमोरी गति माइक्रोप्रोसेसर की गति से काफी धीमी होती है और इन मेमोरी के साथ कार्य करने पर काफी कठनाई होती है।

3.2 मेमोरी के प्रकार

सामान्य रूप से मेमोरी दो प्रकार की होती है।

(1) केवल पढने वाली मेमोरी (Memory only for Read)

(2) पढने तथा लिखने वाली मेमोरी (Memory for Read / Write)

केवल पढने वाली मेमोरी (Memory only for Read,ROM)

इस तरह की मेमोरी को आगे तीन भागों में बाँटा जा सकता है

(a) ROM (Read only Memory)

(b)PROM (Programable Read only Memory)

(c)EPROM (Erasable programmable Read only Memory)

ROM (Read only Memory):-

इस तरह की अर्द्धचालक मेमोरी मे एक बार ही लिखा जाता है उसे किसी भी प्रकार से हटाया नहीं जा सकता है।

प्रत्येक कम्पनी जो कम्प्यूटर बनाती है जब कम्प्यूटर ऑन किया जाता है तो उस कम्प्यूटर के बारे में, उसकी गति के बारे में उसके monitor तथा उसकी मेमोरी जगहों के बारे में जानकारी देती है यह सभी ROM प्रकार की मेमोरी चिप में स्टोर होता है और इसे मिटाया नहीं जा सकता है अर्थात् ROM केवल माइक्रोप्रोसेसर द्वारा पढ़ी जाती है।

(b) PROM (Programable Read only Memory):

यह प्रोग्रामेबल Read only memory को केवल एक ही बार प्रोग्राम किया जाता है। प्रोग्राम हो जाने के बाद यह एक ROM की तरह हो जाती है।

(c) EPROM (Erasable programmable Read only Memory):

इस तरह की मेमोरी में प्रोग्राम को किरणों की सहायता से उसमें लिए प्रोग्रामों को खत्म किया जा सकता है तथा उसे पुनः प्रोग्राम के लिखने लायक बनाय जाता है इस प्रकार की मेमोरी को EPROM कहते हैं। किसी भी मेमोरी में प्रोग्रामों को हटाने में लगने वाला समय उसमें पड़ने वाली किरणों की तीव्रता तथा लगने वा वाले समय पर निर्भर करता है।

(2) पढ़ने तथा लिखने वाली मेमोरी (Memory for Read/ Write): -

कुछ ऐसी मेमोरी चिप भी होती हैं जिनको पढ़ा भी जा सकता है और जिन पर लिखा भी जा सकता है। ऐसी मेमोरी RAM (Randamn Access memory) कहलाती हैं।

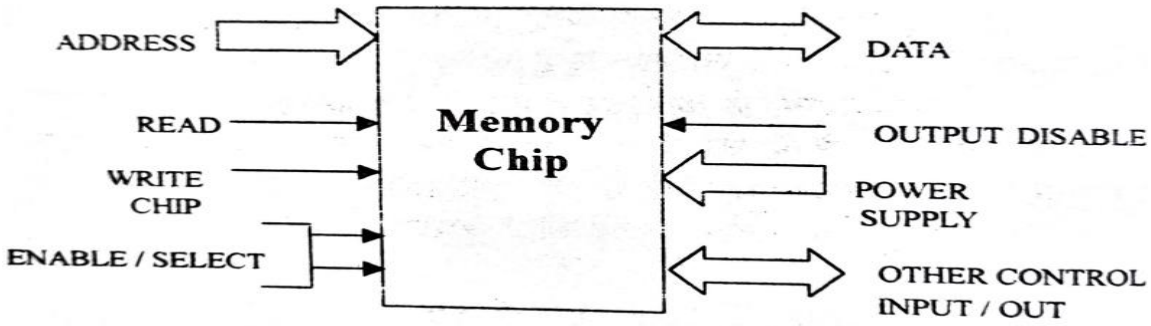
RAM दो प्रकार के होते हैं:

1- स्थैतिक RAM (Static RAM)

2- शक्तियुक्त RAM (Dynamic RAM)

स्थैतिक रेम (RAM) का प्रयोग उन मेमोरी तंत्रो (System) साथ किया जाता है जिनकी साइज छोटी होती है तथा शक्तियुक्त RAM का प्रयोग उन मे के साथ किया जाता है जो बडी होती हैं आजकल के कम्प्यूटर के साथ स्थै RAM का इस्तेमाल कम होता है क्योंकि ये माइक्रोप्रोसेसर की गति के साथ मेल खाती हैं इसलिये शक्तियुक्त RAM का इस्तेमाल ज्यादा किया जाता है।

3.3 मेमोरी चिप की बनावट (Architecture of Memory Chip)



चित्र (3.1) मेमोरी चिप का ब्लाक चित्र

एक मेमोरी चिप में कई महत्वपूर्ण सिग्नल होते हैं जैसा कि चित्र (1) में दिखाया गया है।

एड्रेस बस (Address Bus):-

जैसा कि हम जानते हैं कि अर्द्धचालक मेमोरी को शब्दों में व्यवस्थित किया जाता है। मेमोरी चिप में A_0 से लेकर A_{11} तक 12 लाइन की एड्रेस पिन होती है जिनकी सहायता से मेमोरी चिप को एड्रेस दिया जाता है। यहाँ Ao सबसे छोटी अर्थपूर्ण बिट (Least significant bit) तथा 11 सबसे बडी अर्थपूर्ण बिट (Most significant bit) होती है। इस तरह यह 12 एड्रेस बिट लाइन एड्रेस बस (Address Bus) कहलाती है। उदाहरण के लिये यदि इन 12 एड्रेस लाइनों पर 1011 0101 0011 (AO-A11) दिया जाता है तो हैक्साडेसीमल में $(353)_{16}$ एड्रेस मेमोरी में चला जाता है।

डाटा बस (Data Bus) :

इन मेमोरी चिप का आऊटपुट 8 बिट या 1 बाइट का ही होता है. D_o -D_7 यहाँ पर D_0 सबसे छोटी अर्थपूर्ण बिट (Least significant bit) तथा D_7, सबसे बडी अर्थपूर्ण बिट (Most significant bit) होती है।

READ और WRITE प्रणाली किसी भी निर्देश का हल करने के लिये प्रणाली कोड डाटा बस के जरिये उस मेमोरी जगह जो RAM या ROM में होती है। उनमें चला जाता है। ये माइक्रोप्रोसेसर में वाहक रूप में होते हैं। और इसके पश्चात वह निर्देश हल होता है। इस तरह यह देखा जाता है कि किसी निर्देश को हल करने के लिये दो पदों का इस्तेमाल किया जाता है पहले निर्देश को निकालना (Instruction Fetch) तथा फिर उसे हल करना (Instruction execute)

एक मेमोरी चिप में READ तथा WRITE लाइनें होती हैं जिनका उपयोग मेमोरी चिप से पढ़ने तथा उसमें लिखने का कार्य करने के लिए किया जाता है ।

एक READ प्रणाली को सम्पादित करने के लिये निम्न पदों का इस्तेमाल किया जाता है।

(a) मेमोरी चिप को इस कार्य के लिये Select किया जाता है। मेमोरी चिप मे Chip enable/ select लाइन होती हैं।

(b) एड्रेस इनपुट लाइन पर शब्द के एड्रेस पर रखा जाता है।

(c) READ लाइन को सेट करते हैं जिससे यह पुष्टी हो कि इस समय READ प्रणाली पर कार्य हो रहा है।

(d) एक छोटे से समय अन्तराल (time delay) में जब मेमोरी चिप शब्दों एड्रेस से सूचनाओं को पढ़ लें, डाटा, डाटा आउटपुट लाइन पर आ जाता है। इसी तरह के पद WRITE प्रणाली में भी लिये जाते हैं।

चिप इनेबल / स्लिक्ट लाइन : (Chip enable/select line)

मेमोरी चिप से किसी कार्य को मुख्य रूप से प्रणाली को सम्पादित करने के लिये सर्वप्रथम चिप को तैयार करना पड़ता है कि वह इन प्रणाली पर कार्य करने के लिये तैयार है कि

नहीं। इसके लिये चिप में इनेबल / सिलेक्ट लाइन होती है जिस पर चिप के तैयार होने वाले सिग्नल उत्पन्न होते हैं। कई चिप ऐसी होती हैं जिन पर दो चिप स्लिक्ट लाइन होती हैं।

आउटपुट अक्षम (Output disable):

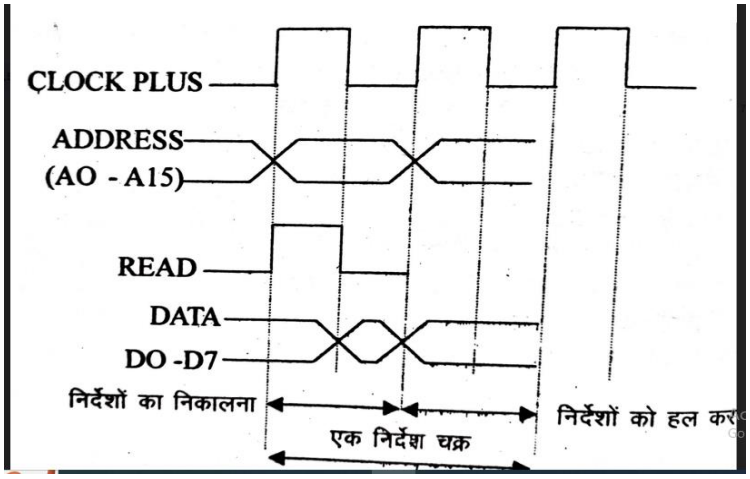
जब मेमोरी चिप पर कोई प्रणाली चल रही होती है तो आऊट पुट डिसेबिल सिग्नल उत्पन्न होते हैं जो बाहर से आने वाले अन्य निर्देशों को रोक देते हैं।

पॉवर सप्लाई (Power Supply):

किसी भी मेमोरी चिप के लिए ये एक अत्यन्त आवश्यक बिन्दु होता है। 5V का सप्लाई पर काम करती है। प्रणाली कार्य के दौरान मेमोरी चिप अपनी अधिकतम सामर्थ्य को दर्शाती है। परन्तु जब मेमोरी चिप पर कार्य नहीं हो रहा हो तो अपनी सामर्थ्य को कम कर लेती है। जिससे इनके आन्तरिक परिपथ में खराबी नहीं आती है।

3.4 READ और WRITE समयचक्र (READ and WRITE Time Cycle):

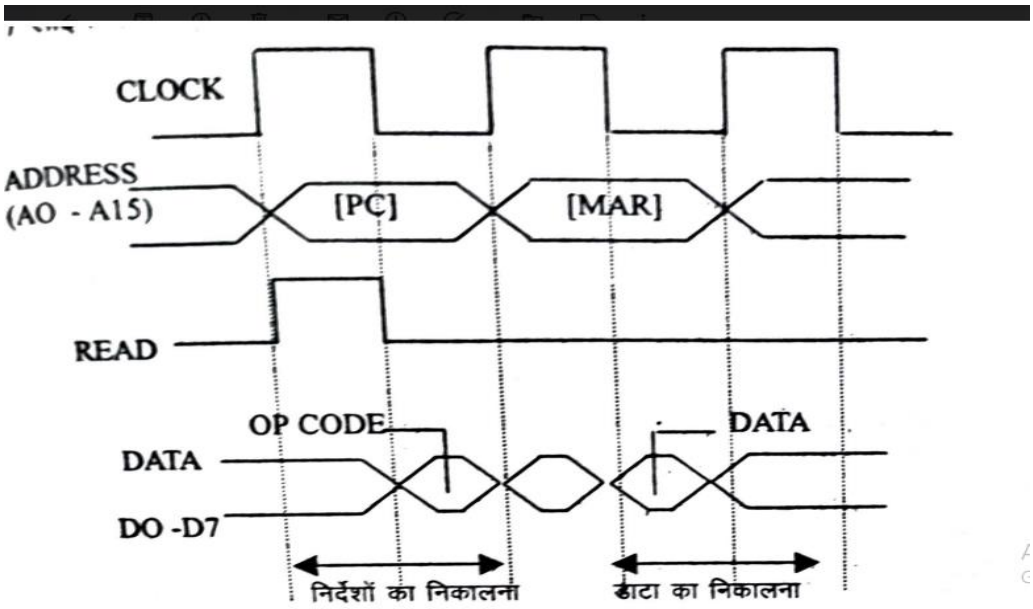
जैसा कि हम जानते हैं कि एक निर्देश एक पूर्ण चक्र में हल होता है य सम्पादित होता है। जैसा कि चित्र (3.2) में दिखाया गया है कि निर्देशों को निकालने की दशा में क्लॉक पल्स high होनी चाहिये। निर्देश (AO-A15) लाइन पर माइक्रोप्रोसेसर के प्रोग्राम कॉउन्टर पर जायेगें। इस समय एड्रेस लाइन उच्च स्तर या कम स्तर हो सकती है। जैसे ही READ पल्स कम स्तर होती है माइक्रोप्रोसेसर एड्रेस लाइन से सूचियों (Contents) को डाटा लाइन DO-D7 पर रख देता है इसके पश्चात् ये माइक्रोप्रोसेसर के रजिस्टर्स में स्टोर हो जाते हैं जहाँ इनकी व्याख्या की जाती है। यह निर्देशों को निकालने की एक विधि है।



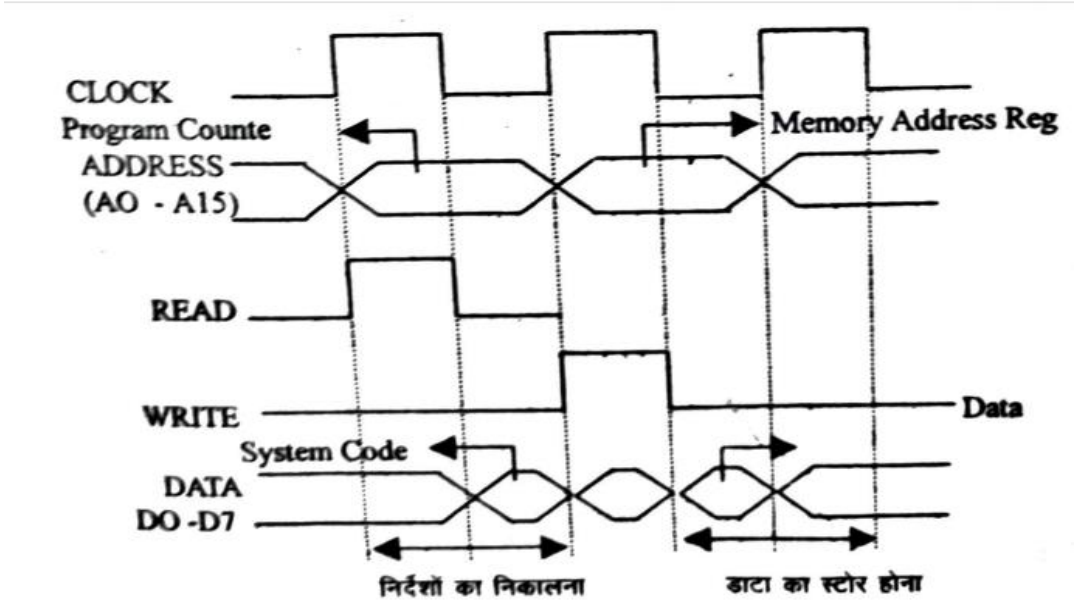
चित्र (3.2) एड्रेस को निकालने का समय चक्र

READ समय चक्र (READ Timing Cycle):

चित्र (3.3) के अनुसार जब क्लॉक पल्स High होती है एड्रेस (Ao-A15) प्रोग्राम काउन्टर (PC) पर जाता है जिसमें एड्रेस रजिस्टर होते जो डाटा काउन्टर कहलाते हैं। जब READ लाइन low होती है तो प्रणाली कोड निकलता है। इसके पश्चात् READ Line दूसरी पल्स में low होती है तो डाटा DO- D7 लाइन में आ जाता है और पढ़ा जाता है



चित्र (3.3) एड्रेस को निकालने का समय चक्र



चित्र (3.4) Write समय चक्र

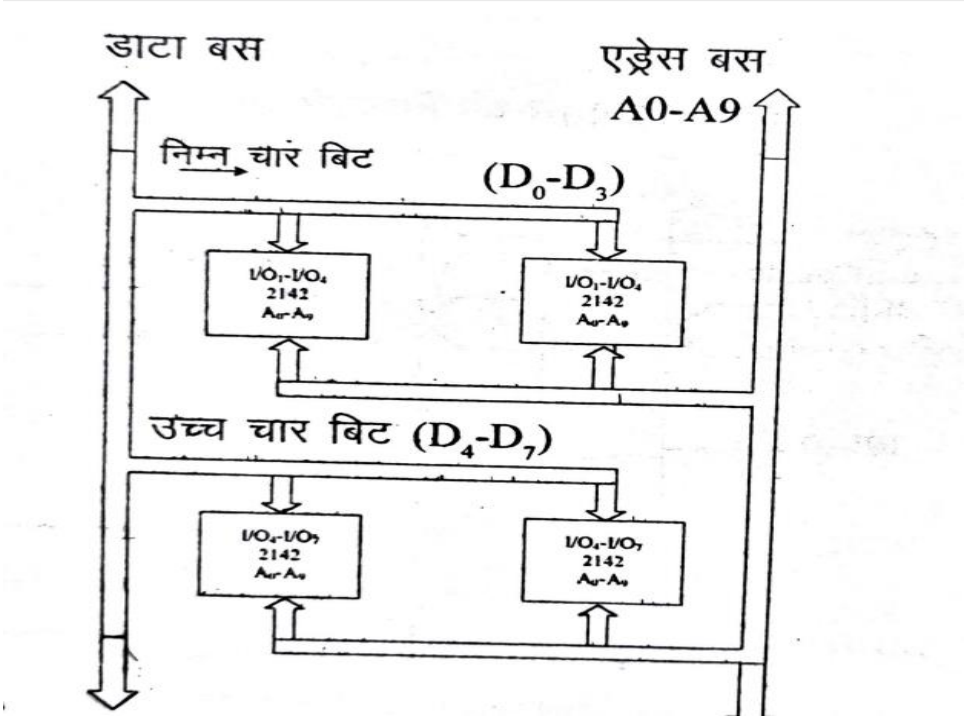
WRITE समय चक्र (WRITE Timing Cycle)

चित्र (3.4) के अनुसार जब READ संकेत सवू होता है तो | कोड से निर्देश निकलते हैं तथा जब WRITE संकेत होते हैं तो डाटा स्टोर हो जात है।

3.5 मेमोरी तंत्र की बनावट (Memory System Organization)

माइक्रोप्रोसेसर आधारित जितने भी तंत्र चलाये जाते हैं सभी में मेमो की आवश्यकता होती है संरचना और आकार के अनुसार इन RAM तथा ROM मेमोरी को जोडना पडता है। उदाहरण के लिये यदि हमें 4 K बाइट का स्थैतिक RAM चाहिये तो हमे चार 2142 चिप को जोडना पडेगा जैसा चित्र (3.5) में दर्शाया गया है। इसमें DO-D7 डाटा बस है जो मेमोरी के जुड़ने के क्रम में निम्न तथा उच्च चार बिट का डाटा भेजता है। A0-A10 एड्रेस बस होती है । प्रत्येक A_0-A_9 , चिप एड्रेस को प्राप्त करती

है तथा चिप पर सिलेक्ट संकेत उत्पन्न किये जाते है जब एड्रेस की दसवीं लाइन उच्च स्तर पर होती है तथा यह प्रदर्शित करते है कि चिप 1 तथा 2 सिलेक्ट की गई है।



चित्र (3.5) मेमोरी चिप 2142 के साथ बनावट

4. 8085 के समय और नियंत्रण इकाई

4.1 परिचय (Introduction)

हम जानते हैं कि माइक्रोप्रोसेसर के साथ अन्य युक्तियाँ भी जुड़ी होती हैं। माइक्रोप्रोसेसर एक कम्प्यूटर में केन्द्रीय प्रोसेसिंग इकाई होती है और इसके साथ इनपुट तथा आऊटपुट युक्तियों, अर्द्धचालक मेमोरी तथा प्रिन्टर इत्यादि जुड़ी होती हैं। इसके अलावा माइक्रोप्रोसेसर के साथ सामान्य उपयोगी रजिस्टर, समय तथा नियंत्रण इकाई तथा डाटा को भेजने वाले रास्ते बस के रूप में होते है। इनमें समय तथा नियंत्रण इकाई का उपयोग काफी महत्वपूर्ण होता है क्योंकि माइक्रोप्रोसेसर एड्रेस को निकालता है और फिर उसे सम्पादित करता है।

4.2 समय तथा नियंत्रण इकाई (Timing & Control Unit)

प्रोग्राम, निर्देशों का एक क्रम होता है जो मेमोरी में संग्रह होते है। यह निर्देश माइक्रोप्रोसेसर के साथ अन्तरमुख होते हैं। माइक्रोप्रोसेसर अपने कार्य को दो भागों में बाटता है पहला निर्देशों को निकालना तथा फिर इन निर्देशों को सम्पादित करना।

पूर्ण निर्देश चक्र = निर्देशों को निकालना + निर्देशों को सम्पादित करना

Instruction Cycle = Fetch Instruction + Execute Instruction

निर्देशों को निकालना फिर इन्हें सम्पादित करना एक निश्चित समय सीमा (Fixed time period) में होते हैं अतः इन्हें निर्देशों को निकालने वाला चक्र (Fetch

Instruction Cycle) सामान्य रूप में निकालन का चक्र (Fetch Cycle) तथा सम्पादित चक्र (Execution Cycle) का योग होता है।

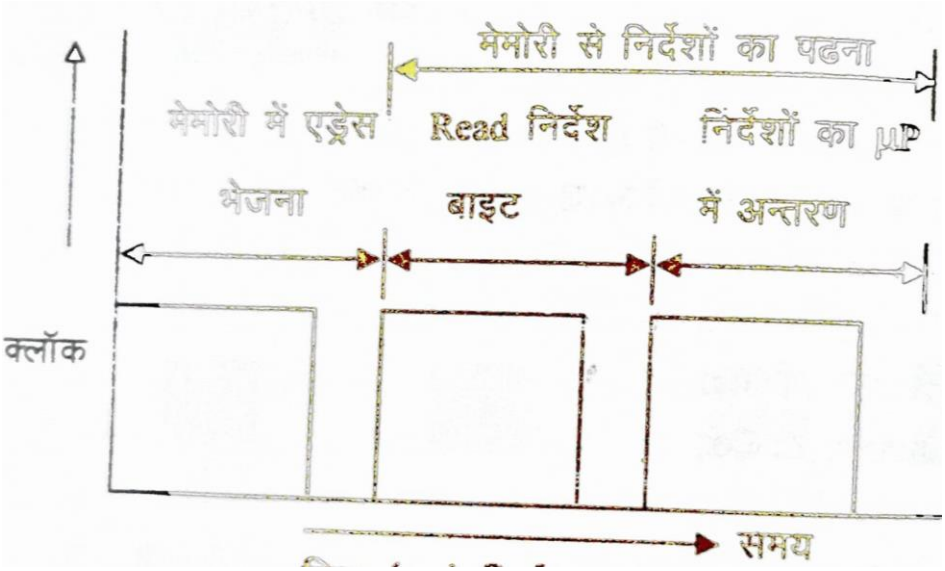
निर्देश चक्र = निर्गत चक्र + सम्पादित चक्र

Instruction Cycle = Fetch Cycle + Execution Cycle

4.2.1. निर्गत चक्र (Fetch Cycle)

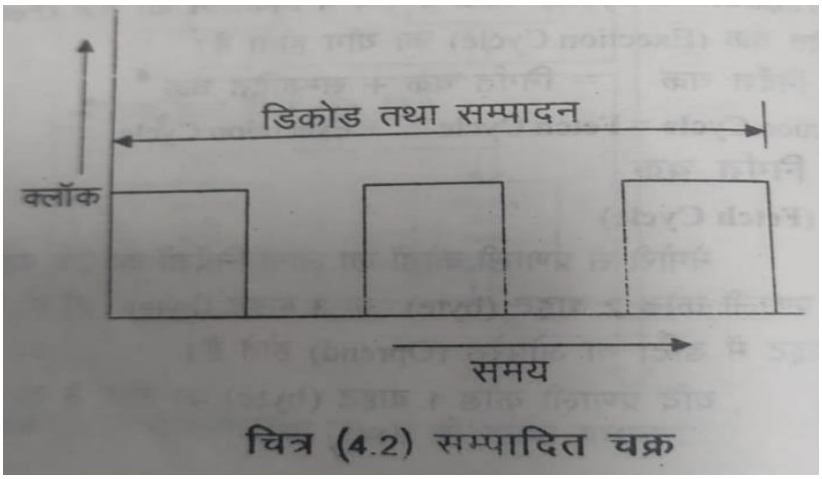
मेमोरी से प्रणाली कोडों का लाना निर्देशों का एक महत्वपूर्ण कार्य हो है। यह प्रणाली कोड 2 बाइट (byte) या 3 बाइट (byte) की हो सकती हैं जिन दूसरी बाइट में डाटा या ओपरेंड (Operand) होते है।

यदि प्रणाली कोड 1 बाइट (byte) का होता है तो भी माइक्रोप्रोस (5.0) की इसे एक निर्गत चक्र (Fetch Cycle) में निकालता है। जैसा कि हम जानते माइक्रोप्रोसेसर में 16 बिट का एक प्रोग्राम काउन्टर (Program Counter) होता है। जिसमें वह एड्रेस होता है जिसे सम्पादित (Execute) करना होता है निर्गत क (Fetch Cycle) की शुरूआत में ही एड्रेस प्रोग्राम काउन्टर के द्वारा मेमोरी जगहों भेज दिये जाते है।



चित्र (4.1) निर्गत चक्र

इस तरह निर्गत चक्र (Fetch Cycle) में एड्रेस को मेमोरी में भेजना तथा मेमोरी से निर्देशों को पढना यह दो कार्य होते हैं जैसा चित्र (4.1) में दर्शाया गया है मेमोरी में एड्रेस भेजने की क्रिया एक क्लॉक चक्र में पूर्ण हो जाती है जबकि मेमोरी से निर्देशों को पढने के लिये दो क्लॉक पल्स का इस्तेमाल किया जाता है जिसमें प्रथम क्लॉक पल्स में निर्देश बाइट को पढ़ा जाता है तथा दूसरे चक्र में इन निर्देशों का माइक्रोप्रोसेसर में अन्तरण किया जाता है।



चित्र (4.2) सम्पादित चक्र

4.2.2 सम्पादित चक्र (Execution Cycle):

जैसा कि चित्र 4.2 में दर्शाया गया है कि सम्पादित चक्र दो क्लॉक चक्र में सम्पादित होते हैं। यहाँ जो महत्वपूर्ण कार्य होता है वह निर्देशों को डिकोड करना होला है। यदि निर्देश 2 बाइट या 3 बाइट के होते हैं तो उनका सम्पादन के लिये क्लॉक पल्स की संख्या अधिक हो जाती है।

4.3 8085 मेमोरी रीड मानचित्र (8085 Memory Read Diagram)

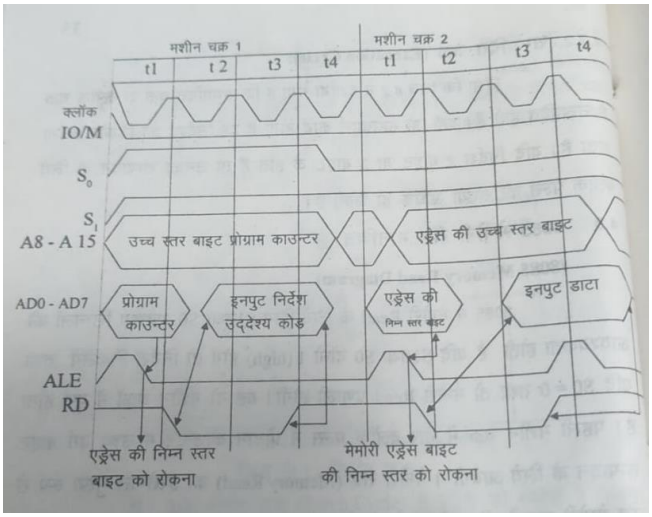
8085 में मेमोरी Read के लिये जिन SIतथा 50 अवस्था सिग्नलों की आवश्यकता होती है यदि SI तथा SO दोनो 1 (high) होगें तो निर्देश निकलेगें तथा • यदि

S0=0 तथा तो मेमोरी Read प्रणाली होगी। यह दो मशीन चक्रों में पूर्ण होता है। पहली मशीन चक्र में चार क्लॉक पल्स में प्रोग्राम काउन्टर में उच्च वर्ग बाइट सम्पादन के लिये आयेगी । मेमोरी रीड (Memory Read) का इस्तेमाल मुख्य रूप से उन मेमोरी जगहों की सूची पढने से होता है जो H-L युग्म रजिस्टर के द्वारा इंगित की जाती है। जैसा कि चित्र (4.3) में दर्शाया गया है कि इस समय IO/M लाइन अर्थात o पर होती है क्योकि इस समय एड्रेस मेमोरी के लिये होते हैं। एड्रेस डाटा बाइट पर प्रोग्राम काउन्टर का निम्न स्तर बाइट आते हैं और इस समय ALE enable होता हैं। यहां प्रथम मशीनी चक्र में S1 तथा S0 दोनों के अवस्था स्तर उच्च होते हैं। और इस समय निर्देशों को निकाला जाता है। दूसरी मशीनी चक्र में यह अवस्था S1 high तथा S0 low होती है और अब मेमोरी से Read किया जाता है। चित्र (4.3) में स्पष्ट है कि जब उच्च स्तर बाइट जाती है तो निम्न स्तर बाइट लेच (latch कर दी जाती है। इसमें संकेत उच्च स्तर पर होते हैं। जब एड्रेस की निम्न स्तर बाइ जाती है तो मेमोरी एड्रेस की निम्न स्तर बाइट को लेच (latch) कर दिया जाता है

4.4 8085 का मेमोरी राइट मानचित्र (8085 Memory write diagram)

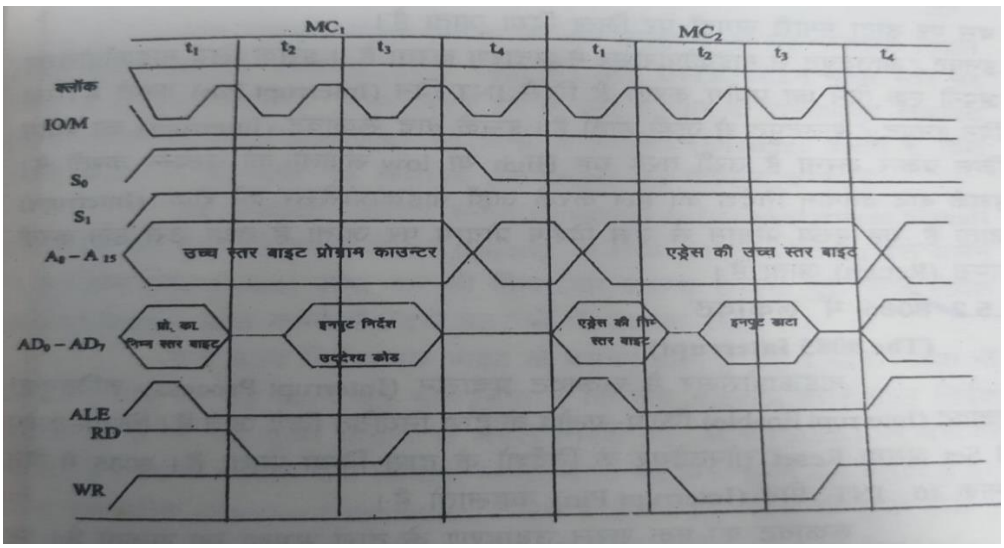
8085 में मेमोरी राइट के लिये दो मशीन चक्रों का इस्तेमाल किया जाता है। प्रथम चार क्लॉक पल्स निर्देशों को निकालने में प्रयुक्त होती है तथा दूसरे चक्र में तीन क्लॉक पल्स में मेमोरी में लिखा जाता है।

इस समय अवस्था संकेत S0 = 1, S1 = 0 होता है तथा WR = 0 इंगित करता है जो मेमोरी राइट (Memory Write) प्रणाली को बताता है। जैसा कि चित्र (4.4)



चित्र (4.3) मेमोरी रीड मानचित्र

में दर्शाया गया है कि प्रथम मशीनी चक्र में निर्देश निकाले जाते है इस समय अवस्था संकेत S1 तथा S0 उच्च स्तर पर होते है तथा IO/M लाइन निम्न स्तर पर होती है। एड्रेस लाइन पर प्रोग्राम काउन्टर (Program Counter) की स्तर बाइट होती है डाटा बस पर एड्रेस की निम्न स्तर बाइट को लेच (letch) किया जाता है। अगले मशीनी चक्र में मेमोरी एड्रेस की निम्न स्तर बाइट को लेच (letch) करते है और डाटा



चित्र (4.4) मेमोरी रीड मानचित्र

4.5 रूकावट (Interrupt)

माइक्रो प्रोसेसर में प्रोग्रामिंग के दौरान कई बार यह देखा जाता है कि मुख्य प्रोग्रामो के साथ कोई एक भाग का इसमें बार-बार प्रयोग होता है। दूसरे शब्दों में, जब वाहन (Carrier) इनपुट / आउटपुट युक्तियों माइक्रोप्रोसेसर को रोक (Interrupt) सकती है क्योंकि रुकावट (Interrupt) की यह पिन माइक्रोप्रोसेसर में मौजूद होती है जब वाह्य युक्ति से इस संकेत को चलाया जाता है तो माइक्रोप्रोसेसर एक विशेष प्रोग्राम में चला जाता है जिसे ISR (Interrupt Service Routine) कहते हैं। इस तरह, एक विशेष प्रोग्राम में जाने के बाद इसे अपने मूल प्रोग्राम में वापस (Return) आना पड़ता है।

4.5.1 इनपुट / आउटपुट रूकावट (Input / Output Interrupt)

एक माइक्रोप्रोसेसर डाटा को इनपुट / आउटपुट मे तथा बस पर डाटा मेमोरी जगहों पर लिख दिया जाता है। इनपुट / आउटपुट से माइक्रोप्रोसेसर मे अन्तरण करता है । इसके लिये माइक्रोप्रोसेस अपनी एक पिन का प्रयोग करता है जिसे INT पिन (Interrupt Pin) कहते हैं। पिन इनपुट / आउटपुट से जुडी होती है। इसके बाद रूकावट (Interrupt) का प्रयो किस प्रकार करना है उसी तरह यह High या low संकेतो को उत्पन्न करती है इसके बाद वर्तमान निर्देश को हल करके जहाँ माइक्रोप्रोसेसर को रोका (Interrup जाता है. यह मुख्य प्रोग्राम से उस विशेष प्रोग्राम पर जाता है तथा उसे हल कर वापस (Return) आता है।

4.5.2 8085 में रूकावट (The 8085 Interrupt)

माइक्रोप्रोसेसर में रूकावट प्रचालन (Interrupt Process) शक्तियु रुकावट (Interrupt Enable) फ्लिप-फ्लोप के द्वारा नियंत्रित किये जाते हैं। फ्लिप फ्ल को Set अथवा Reset सोफ्टवेयर के निर्देशों के द्वारा किया जाता है। 8085 में फि क्रमांक 10 INT पिन (Interrupt Pin) कहलाती हैं।

रूकावट को एक सरल उदाहरण के द्वारा समझा जा सकता है। ह सभी ने ट्रैफिक नियंत्रित संकेतो को देखा है जो अकसर चौराहे पर लगे होते हैं। एक ट्रैफिक संकेत के लिये तीन रंगो के प्रकाश का इस्तेमाल किया जाता है। लाल, पील तथा हरा । इन प्रत्येक रंगो के प्रकाश के लिये यह सुनिश्चित करना पड़ता है कि किस रंग का प्रकाश कितने समय के लिये रखना है अर्थात आपके पास एक-एक प्रोग्राम समय विलम्ब (Time delay) को होना चाहिये जिससे चौराहे का प्रत्येक रा के लिये संकेत व्यवस्था की जा सके। अगर देखा जाये तो संकेत नियंत्रक के लिए माइनोप्रोसेसर को प्रोग्राम किया जा सकता है और मुख्य प्रोग्राम में हर बार इस सम विलम्ब (Time delay) को हल किया जाये या फिर हम इन समय विलम्ब (Time delay) के प्रोग्राम को एक जगह बना कर रख ले तथा मुख्य प्रोग्राम में जब भी इसक आवश्यकता पड़े इसे इन्हें मुख्य प्रोग्राम को रोककर (Interrupt) इन समय विलम (Time delay) प्रोग्राम को हल किया जाये और फिर वापस मुख्य प्रोग्राम में आय जाये। यह रूकावट क्रिया (Interrupt Process) कहलाती है।

8085 में जब रूकावट क्रिया (Interrupt Process) की जाती है सर्वप्रथम इस क्रिया को शक्तियुक्त (Enable) किया जाता है तथा इसके लिये (Enable Interrupt) का प्रयोग किया जाता है। यह निर्देश रूकावट फ्लिप फ्लॉप -शक्तियुक्त (Enable) करता है। यदि फ्लिप फ्लोप को दूसरी बार निर्देशित करना है तो DI (Disable Interrupt) का प्रयोग किया जाता है।

जब माइक्रोप्रोसेसर निर्देशों को हल करता है तो यह रूकावट लाइन (Interrupt Line) को प्रमाणित करता रहता है। यदि INTR लाइन उच्च स्तर पर जाती है तो माइक्रोप्रोसेसर जिस निर्देश पर है उसे हलकर के शक्तियुक्त रुकावट फ्लिप फ्लोप (Enable Interrupt) को बिना शक्तियुक्त (Disable) करता है तथा संकेत भेजता है जो मांगी गई रूकावट (Interrupt) को स्वीकार करता है।

यह रूकावट स्वीकार संकेत (Interrupt Acknowledgement Signal) एक निर्देश को मुख्य प्रोग्राम मे डालने (Insert) के लिये उपयोग होता है। इसे अलग से हार्डवेयर (Hardware) लगा कर भी किया जा सकता है इसे पुनः चालू निर्देश (Restart Instruction) कहते हैं। इसे RST से भी दर्शाया जाता है।

एक RST निर्देश एक बाइट के बराबर होता है इसे CALL निर्देश भी कहते है! यहाँ पर यह ध्यान देने योग्य बात है कि जब माइक्रोप्रोसेसर RST निर्देश प्राप्त करता है तो यह मुख्य प्रोग्राम के अगले निर्देश को स्टेक (Stack) पर बचाकर रख लेते है। इसके पश्चात उस विशिष्ट मेमोरी जगह पर प्रोग्राम को हल करता है इसे सर्विस सबरूटीन (Service Subroutine) कहते हैं इससे EI निर्देश भी जुड़ा होता है जब यहाँ प्रोग्राम पूरी तरह हल हो जाता है तो RET निर्देश (Return Instruction) द्वारा यह वापस मुख्य प्रोग्राम मे वापस आ जाता है जहाँ प्रोग्राम छोड़ा जाता है।

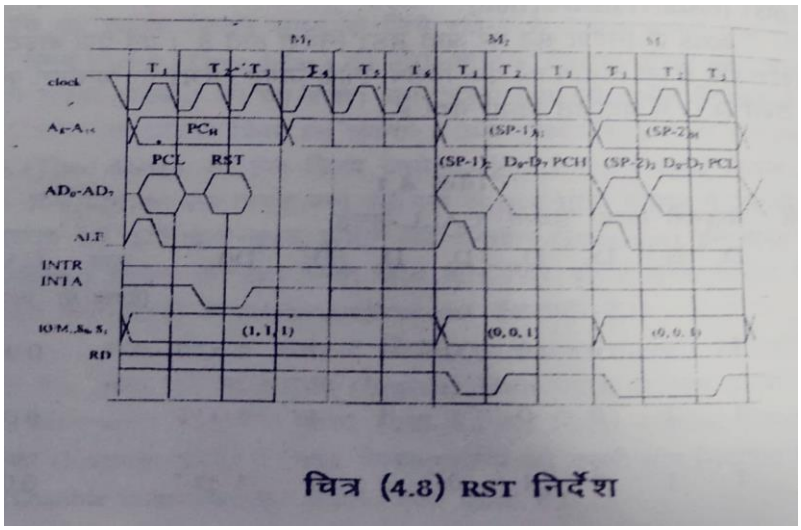
4.5.3 RST निर्देश [RST (restart) Instruction]

8085 के निर्देश सेट पर आठ RST निर्देश होते हैं। यह एक बाइट के CALL निर्देश होते हैं जो प्रोग्राम को एक विशेष मेमोरी जगह OOH पर अन्तरण करते हैं। इसे टेबिल (4.1) में प्रदर्शित किया गया है।

मेनोमोनिक्स	बाइनरी कोड हेक्सा								CALL जगह कोड (हेक्सा में)	CALL जगह
	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0		
RST 0	1	1	0	0	0	1	1	1	1C7	0000
RST 1	1	1	0	0	1	1	1	1	1CF	0008
RST 2	1	1	0	1	0	1	1	1	1D7	0010

RST 3	1	1	0	1	1	1	1	1	1DF	0018
RST 4	1	1	1	0	0	1	1	1	1E7	0020
RST 5	1	1	1	0	1	1	1	1	1EF	0028
RST 6	1	1	1	1	0	1	1	1	1F7	0030
RST 7	1	1	1	1	1	1	1	1	1FF	0038

चित्र (4.5) के अनुसार जब INTR line शक्तियुक्त होती है तो स्वीकार करने वाले संकेत उत्पन्न होते हैं। प्रथम मशीन (M) चक्र में RST निर्देश डाटा बस पर लाया जाता है जो कि स्टेक पर होता है और यही कारण है कि सि रूटीन (Service routine) के बाद भी प्रोग्राम चलता रहता है। M, चक्र में एड्रेस पर स्टेक पाइन्टर एक जगह रखी जाती है और प्रोग्राम काउन्टर में उच्च स्तरी एड्रेस M, चक्र में फिर प्रोग्राम काउन्टर की निम्न स्तर एड्रेस लोड होते है तथा स्टेककी (SP-2) जगह पर रखा जाता है।

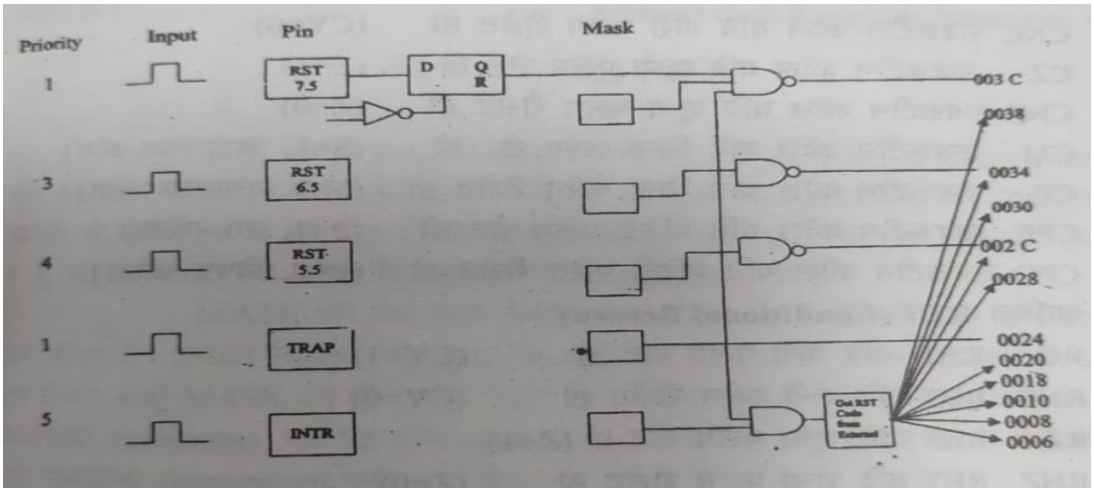


चित्र (4.8) RST निर्देश

4.5 अन्य 8085 रूकावट (Additional 8085 Interrupt)

8085 में पाँच रूकावट (Interrupt) इनपुट होते हैं जिनमें एक INTR (पिन क्रमांक 10) तथा अन्य चार क्रमश: TRAP, RST 7.5, RST 6.5, RST

5.5 (पिन क्रमांक 6 से 9) होते हैं जिनकी विशेष मेमोरी जगहें होती हैं और ये वही अन्तरण होते। इन्हें स्वचालित वेक्टर (Automatically Vectored) भी कहते हैं।



चित्र (4.6) 8085 के रुकावट (Interrupts)

जैसा कि चित्र (4.6) से स्पष्ट है कि 8085 के चार रूकावट (Interrupt) वेक्टर होती हैं और यदि ये की जाती हैं तो प्रोग्राम सीधे निर्देशित मेमोरी जगहों पर चला जाता है। इसके अलावा भी मेमोरी जगहें बनाई जा सकती हैं जो सीधे मुख्य प्राग्राम पर रूकावट (Interrupt) करने के बाद होती है इसके लिये एक वाह्य हार्डवेयर की आवश्यकता होती है जो चित्र 4.6 के अनुसार मेमोरी जगहों को बनाता हैं। सामान्यतः RST 7.5, RST 6.5, RST 5.5 मुखडों सहित रूकावटें (Maskable Interrupt) होते हैं। जहाँ तक TRAP का प्रश्न है ये मुखड़ों वाला नहीं (Non Maskable) होता है इसे NMI भी कहते हैं।

4.7 आश्रित CALL तथा RET निर्देश (Conditional Call and Return instruction)

निराश्रित CALL तथा निराश्रित RET (Return) के अलावा 8085 आठ

आश्रित CALL (Conditional Call) तथा आठ आश्रित RET (Conditional Return) होते है। इन अवस्थाओं में Flag का परीक्षण किया जाता है । आश्रित CALLS के अनुसार जब अवस्था (Condition) मिलती है तब ही प्रोग्राम सबरूटीन (Sub tine) में जाता है अन्यथा नहीं। और जब सबरूटीन (Subroutine) में प्रोग्राम के RET (Return) अवस्था मिलती है तब ही ये वापस मुख्य प्रोग्राम में आते हैं।

आश्रित कॉल (Conditional Call)

CC	सबरूटीन कॉल यदि कैर्री फ्लैग सेट हो	(CY=)
CNC	सबरूटीन कॉल यदि कैर्री फ्लैग रिसेट हो	(CY=0)
CZ	सबरूटीन कॉल यदि शून्य फ्लैग सेट हो	(Z=1)
CNZ	सबरूटीन कॉल यदि शून्य फ्लैग रिसेट हो	(Z=0)
CM	सबरूटीन कॉल यदि चिन्ह फ्लैग सेट हो	(S = 1, ऋणात्मक संख्या)
CP	सबरूटीन कॉल यदि चिह्न फ्लैग रिसेट हो	(S= 0, सकारात्मक संख्या)
CPE	सबरूटीन कॉल यदि पैरिटी फ्लैग सेट हो	(P = 1, सम संख्या)
CPO	सबरूटीन कॉल यदि पैरिटी फ्लैग रिसेट हो	(P = 0, विषम संख्या)

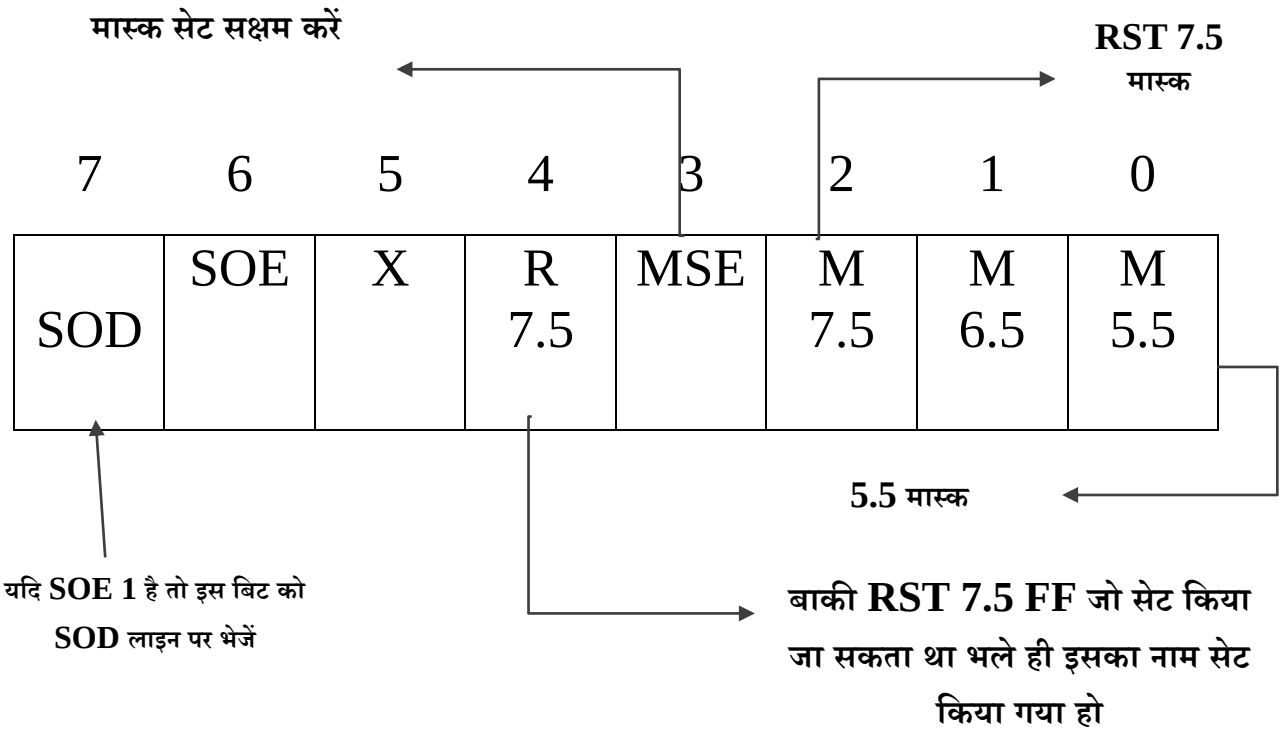
आश्रित RET

RC	RET यदि कैर्री फ्लैग सेट हो	(CY=1)
RNC	RET यदि कैर्री फ्लैग रिसेट हो	(CY=0)
RZ	RET यदि शून्य फ्लैग सेट हो	(Z=1)
RNZ	RET यदि शून्य फ्लैग रिसेट हो	(Z=0)

RM	RET यदि चिन्ह फ्लैग सेट हो	(S = 1, ऋणात्मक संख्या)
RP	RET यदि चिह्न फ्लैग रिसेट हो	(S= 0, सकारात्मक संख्या)
RPE	RET यदि पैरिटी फ्लैग सेट हो	(P = 1, सम संख्या)
RPO	RET यदि पैरिटी फ्लैग रिसेट हो	(P = 0, विषम संख्या)

चित्र (4.7) SIM निर्देश

जैसा कि चित्र 4.7 मे दिखाया गया है यह ACC बिट के समान 8 बिट को MASK करता है। इसलिये आवश्यक यह होता है कि SIM निर्देश को सम्पादित करने से पहले ACC को सेट किया जाना चाहिये। यदि ACC मे तुल्य MASK बिट 1 है तो MASK सेट होता है नहीं तो रीसेट (Reset) होती है।

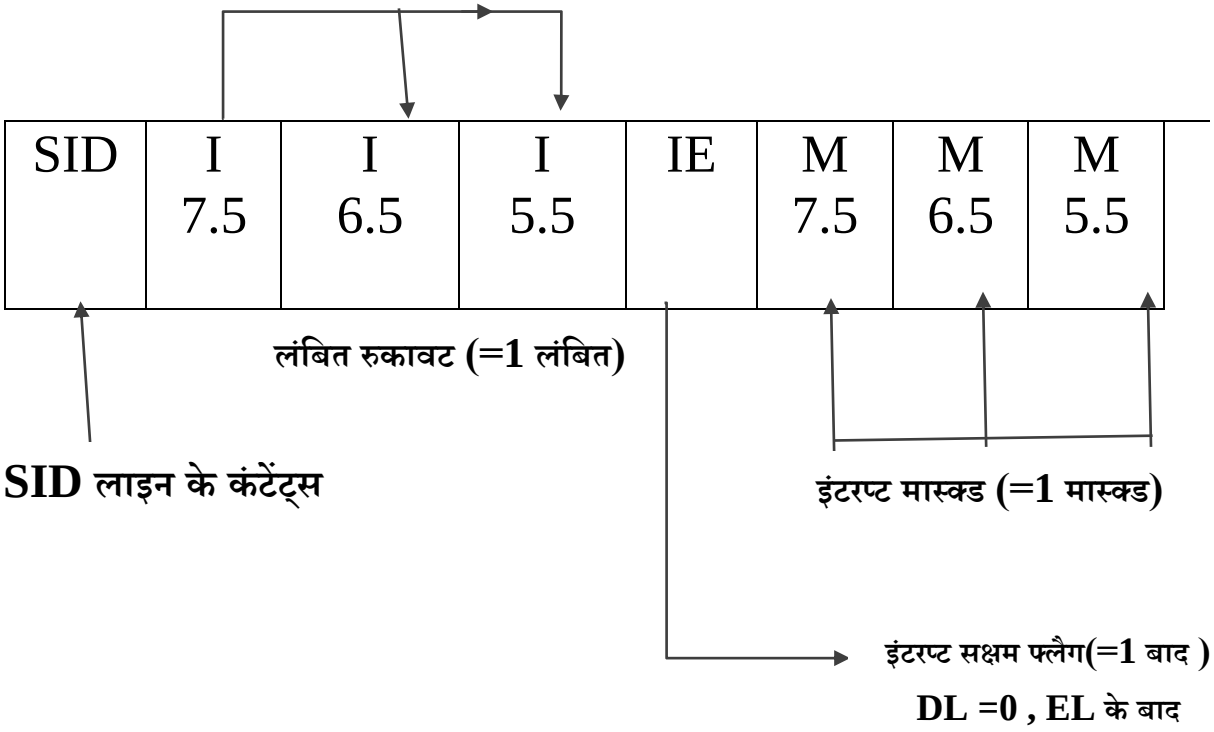


MASK का सेट तथा रीसेट होना (चित्र 4.7 में तीसरी बिट) के 1 होने पर होता है। इसका मतलब यह हुआ कि यदि हमे दो निर्देशों की परीक्षा करनी पड़े तो (चित्र 4.9) MASK का सेट और रीसेट होना आवश्यक है यदि MASK सेट है तो रुकावट (interrupt) को पहचाना (recognize) गया है। यदि रीसेट है तो रूकावट को पहचाना (recognize) नहीं गया है।

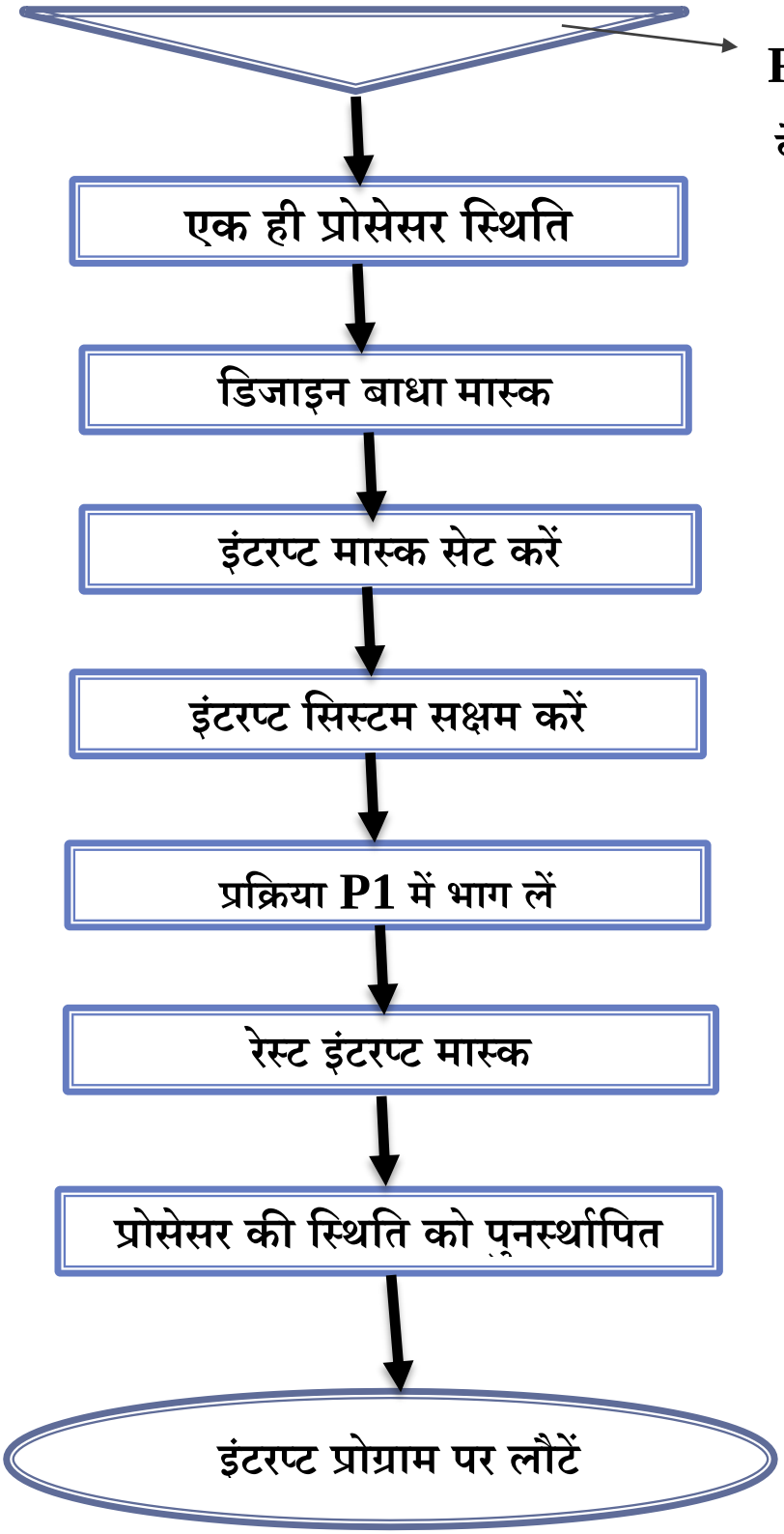
4.9 रीड रूकावट निर्देश (Read Interrupt Instruction)

हर माइक्रोप्रोसेसर, MASK को संतुलन प्रदान करता है और इसी के साथ यह एक निर्देश प्रदान करता है जिसकी सहायता से इन MASK को पढ़ा जा सकता है। 8085 में RIM निर्देश जब उपयोग में आता है तो ACC चित्र (4.8) के अनुसार स्थापित हो जाता है ।

इस तरह की उपयोगिता माइक्रोप्रोसेसर पर काम करने वालों को होती है जो RIM निर्देशों को पढकर उन्हें आगे कैसे करना है इसका निर्धारण करते है। वर्तमान में वेक्टर निर्देशित रूकावटों (interrupt) का अधिक उपयोग होने की वजह से SIM तथा निर्देशों का कम इस्तेमाल होता है। क्योंकि वेक्टर रूकावट (Vector interrupt) में हम सीधे ही उस मेमोरी जगह को प्रभावित कर देते हैं।



चित्र (4.8) RIM निर्देश



चित्र (4. 9) दो निर्देशों की परीक्षा के लिए प्रोग्राम

5. एड्रेस और डाटा स्थानांतरण का वर्गीकरण

5.1 परिचय (Introduction)

एक माइक्रोप्रोसेसर या तो मेमोरी या फिर इनपुट / आउटपुट के साथ अन्तरमुख (Interface) होता है। जब मेमोरी के साथ अन्तरमुख (Interface) होता है। तो इसका उपयोग डाटा को रखने में या फिर पढ़ने में होता है। जब माइक्रोप्रोसेसर, इनपुट / आउटपुट के साथ अन्तरमुख (Interface) होता है तो वातावरण के साथ व्यवहार करता है। अगर दूसरी तरफ से देखा जाये तो माइक्रोप्रोसेसर से डाटा का अन्तरण मेमोरी में या फिर इनपुट / आउटपुट के साथ होता है। माइक्रोप्रोसेसर के साथ डाटा का अन्तरण समय पर निर्भर करता है यह इस बात पर भी निर्भर करता है कि डाटा किस खाके (Format) में अन्तरित हुआ है । जब दो युक्तियां अन्तरमुखी (Interfaced) होती हैं तो उनके विद्युत अभिलाक्षणिक के बारे में भी जानना जरूरी होता हैं।

अन्तरमुख में समय का यह अन्तर धीमी गति की मेमोरी अथवा इनपुट/आउटपुट युक्तियों के कारण होता है। डाटा अन्तरण के खाके (Format) पर भी यह विभिन्नता दिखाई देती है। जैसे एनॉलाग तंत्र (Analog System) में डाटा एक क्रम में वितरीत होता है वही माइक्रोप्रोसेसर में यह समानान्तर क्रम में लिया जाता है। जो कि स्त्रोत से एक प्रकार की विभिन्नता उत्पन्न करते हैं।

5.2 एड्रेस जगहों खाने (Address Space Partitioning)

एड्रेस जगहों से तात्पर्य, एड्रेसो के सेट से होता है । माइक्रोप्रोसेसर के द्वारा आये एड्रेस अथवा इनपुट / आउटपुट युक्तियाँ द्वारा आये एड्रेस को रखने के लिये मेमोरी चिप में जगह

बाँटी जाती है। जैसा कि हम जानते हैं कि इनपुट/आउटपुट की एड्रेसिंग (Addressing) दो तरह की योजनाओं के साथ की जाती है। पहली मेमो मेप्ड इनपुट / आउटपुट योजना तथा दूसरी इनपुट / आउटपुट मेप्ड इनपुट / आउटपुट योजना। अगर इन योजनाओं का इस्तेमाल किया जाना है तो यदि 64 K (K=1024) बाइट की जगह में मेमोरी मेप्ड इनपुट / आउटपुट योजना के तहत हमे EPROMRAM तथा I/O के लिये जगह छोड़नी पड़ेगी जैसा चित्र (5.1) में दर्शाया गया है

FFFF

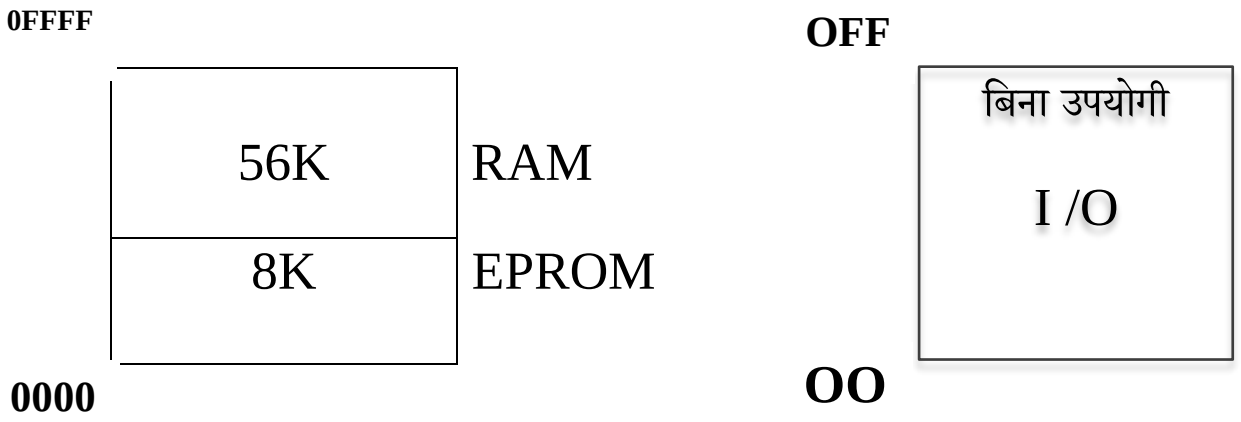
8K	I/O
48K	RAM
8K	EPROM

0000

चित्र (5.1) - 64 K एड्रेस जगह

(मेमोरी मेप्ड I/O योजना के तत्व)

इसी तरह I/O मेप्ड I/O योजना के तहत माइक्रोप्रोसेसर दो तरह की सुविधा प्रदान करता है एक तो सीधे एड्रेस को दिया जा सकता है या फिर एड्रेस को भेजा जा सकता है। जैसा कि चित्र (5.2) में दर्शाया गया है।



चित्र (5.2) - 64K एड्रेस जगह

(I/O मेपड I/O योजना के तहत)

5.3 मेमोरी अन्तरमुख (Memory Interfacing)

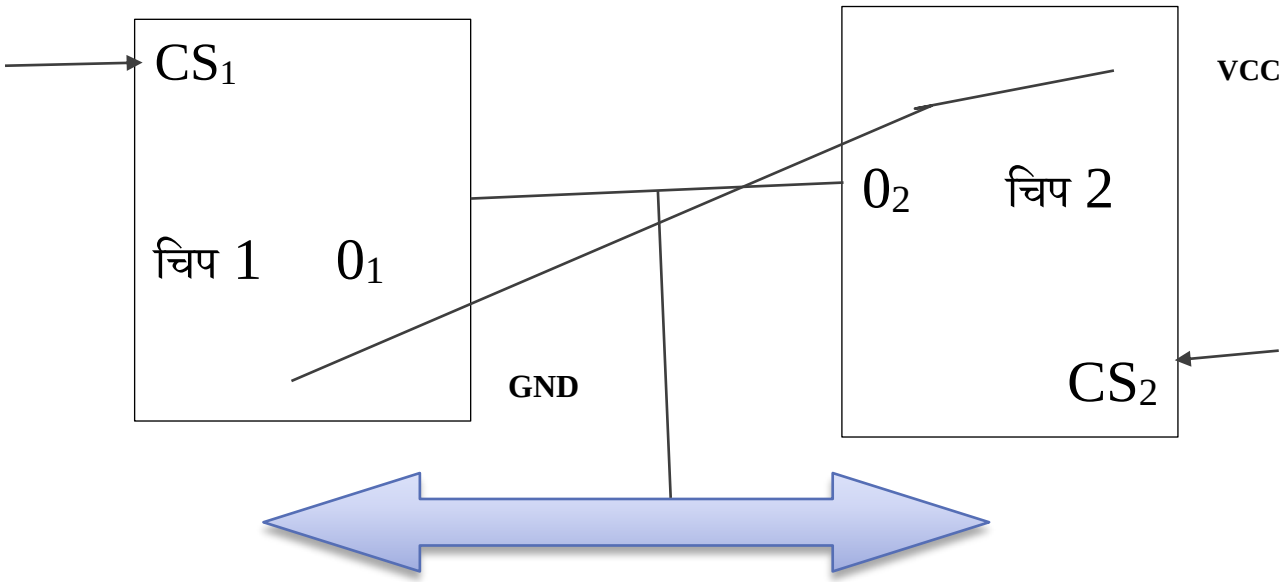
माइक्रोप्रोसेसर के साथ मेमोरी को अन्तरमुख (Memory Interfacing) करने के लिये हमें मेमोरी चिप को इस तरह चुनना पड़ता है कि उसकी माइक्रोप्रोसेसर के साथ विभिन्नता न हो। मेमोरी का माइक्रोप्रोसेसर के साथ अन्तरमुख (Interface) होने के लिये इसकी गति तथा इसके पढने की क्षमता माइक्रोप्रोसेसर की गति के साथ मिलनी चाहिये।

बस विवाद तथा 2- लाइन नियंत्रण (Bus Contention and 2-line Control)

चित्र (5.3) के अनुसार दो मेमोरी चिप पिन के आउटपुट एक ही तंत्र डाटा बस के साथ एक लाइन नियंत्रक OR-गेट के साथ जुड़ी है।

जब पहली चिप चुनी जाती है तो रीड प्रचालन के लिये CS लाइन निम्न स्तर पर होती है। और एक समय देरी (time delay) के पश्चात वापस सामान्य स्थिति में आ जाती है और माइक्रोप्रोसेसर के द्वारा डाटा पढ लिया जाता है इसके पश्चात पहली चिप का CS

लाइन उच्च स्तर पर चली जाती है। इसके पश्चात एड्रेस दूसरी चिप पर हस्तान्तरित होते है तथा दूसरी चिप का CS लाइन निम्न स्तर पर जाती है तथा ऊपर दी गई क्रिया दुबारा होती है।

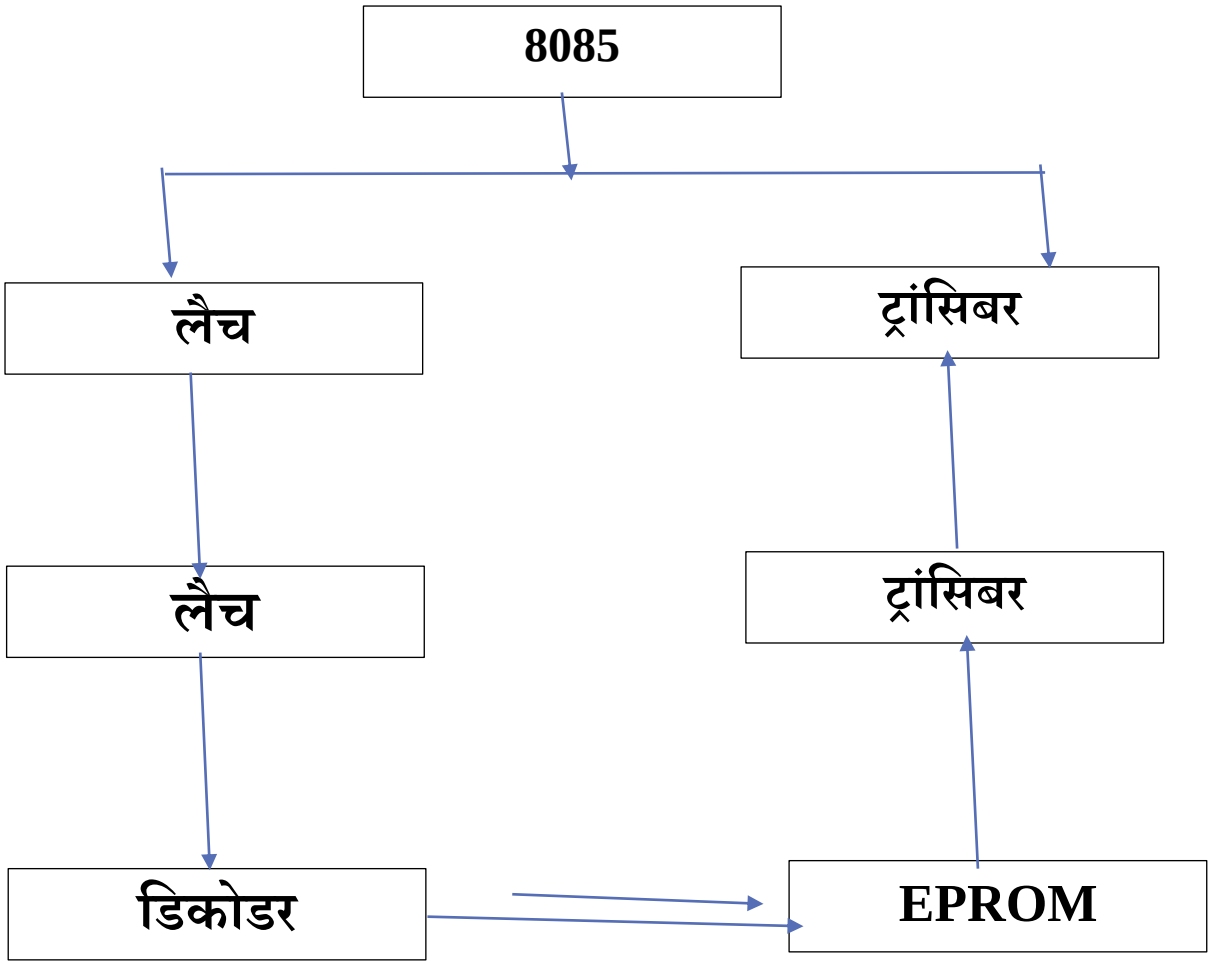


चित्र (5.3) एक लाइन नियंत्रण

5.4 अधिक समय की गणना (Access Time Computation)

जब माइक्रोप्रोसेसर के साथ मेमोरी चिप अन्तरमुख (Interface) होती है तो अगर इनकी गति माइक्रोप्रोसेसर से समानता न रखे तो प्रचालन के दौरान एक समय देरी उत्पन्न होती है।

हम एक ब्लॉक चित्र (5.4)पर विचार करते है जहा पर समय देरी के कई स्तर मौजूद हैं।



चित्र (5.4) तंत्र जिसमें समय देरी के कई स्तर हों

जैसा कि चित्र (54) से स्पष्ट है कि 8085 से डाटा जब जाता है तो लगभग 600 μs के बाद लौटता है एड्रेस के प्रसारण के बाद। जब EPROM से डाटा निकलता है तो यहाँ समय देरी (access time) शून्य होता है और डाटा जब माइक्रोप्रोसेसर तक पहुँचता है तो यह समय देरी लगभग 550 Ls की हो जाती है इसके मध्य डाटा को आगे बढाने के लिये चालक लगे होते हैं जो लगभग 30 JIS का समय लेते हैं। यही कारण है कि READ तथा WRITE प्रचालन के दौरान माइक्रोप्रोसेसर HOLD अवस्था मे जाता है क्योंकि यह विभिन्नता को स्पष्ट करना जरूरी है। चित्र (5.4) में एकल पट्टा (Single Band) मे विभिन्नता अधिक होती है जबकि देरी इनमें ज्यादा इनमें ज्यादा होती है क्योंकि देरी उत्पन्न करने वाले स्तर ज्यादा होते हैं।

5.5 डाटा अन्तरण योजना (Data Transfer Scheme)

डाटा अन्तरण माइक्रोप्रोसेसर में तीन प्रकार से होता है :-

1. माइक्रोप्रोसेसर से मेमोरी
2. माइक्रोप्रोसेसर से इनपुट / आउटपुट युक्तियों में
3. मेमोरी से इनपुट / आउटपुट युक्तियों में

यहाँ प्रत्येक युक्तियों का काम डाटा को निश्चित समय में पाना या उसे एक निश्चित समय में भेजना होता है। इस प्रक्रिया के दौरान माइक्रोप्रोसेसर की विभिन्नता युक्तियों के साथ हो जाती है। उदाहरण के लिये मेमोरी तथा इनपुट / आउटपुट युक्तियों की गति माइक्रोप्रोसेसर की गति से धीमी हो सकती है। जब यह युक्तियां माइक्रोप्रोसेसर के साथ अन्तरमुख (Interface) होती हैं तो मेमोरी या युक्ति माइक्रोप्रोसेसर की गति से धीमी होती है और माइक्रोप्रोसेसर होल्ड (Hold) अवस्था मे जाता है और यदि मेमोरी तथा I/O युक्ति तेज गति की होती है तो इनसे आये डाटा को रोकने के यंत्र (buffer) में डाला जाता है जहाँ से डाटा माइक्रोप्रोसेसर के द्वारा बाद में इकट्ठा कर लिया जाता है।

5.5.1 वर्गीकरण (Classification)

माइक्रोप्रोसेसर तथा I/O युक्तियों के बीच डाटा का अन्तरणएक योजना के तहत किया जाता है जो निम्न है

1. प्रोग्राम डाटा अन्तरण (Programmed data transfer)

2. सीधे मेमोरी अधिक देरी अन्तरण (Direct Memory Access transfer)

5.5.2 प्रोग्राम डाटा अन्तरण (Programmed data transfer)

इसमें एक घरेलू मेमोरी रूटीन का उपयोग किया जाता है जो युक्ति को माइक्रोप्रोसेसर के किसी एक रजिस्टर्स से डाटा अन्तरण किया जाता है। इसका उपयोग बहुत कम डाटा के अन्तरण के लिये किया जाता है। इन्हें मुख्य रूप तीन भागों में बांटा जाता है :-

(1) एक साथ होने वाला अन्तरण (Synchronous transfer)
(2) एक साथ न होने वाला अन्तरण (Asynchronous transfer)
(3) रूकावट के द्वारा चलाया गया अन्तरण (Interrupt driven transfer)

(1) एक साथ होने वाला अन्तरण (Synchronous transfer)

किसी भी डाटा अन्तरण योजना कि यह सबसे सरल विधि है। यह अन्तरण तभी संभव है जब माइक्रोप्रोसेसर तथा 10 युक्ति की (Speed) समान हो।

जब किसी युक्ति से या युक्ति (device) को डाटा अन्तरण किया है तो उपयोग करने वाले प्रोग्राम में एक अनुरूप निदेश युक्ति दी जाती है।उदाहरण के लिये MOV तथा OUT इत्यादि का उपयोग एक साथ होने वाले अन्तरण में किए जाता है।

यह अन्तरण निम्न पदों में सम्पादित होता है।

युक्ति को तैयार रहने की प्रार्थना

और जब तक युक्ति तैयार होती है कोई और कार्य करना

इनपुट / आऊटपुट निर्देश उत्पन्न करना

(2) एक साथ न होने वाला अन्तरण (Asynchronous transfer)

इस तरह का अन्तरण उस समय उपयोग में लाया जाता है जब माइक्रोप्रोसेसर तथा इनपुट / आऊटपुट युक्ति की चाल Speed एक दूसरे के साथ नहीं मिलती है। इस योजना के तहत माइक्रोप्रोसेसर एक निर्देश देता है और तब तक इंतजार करता है जब तक युक्तियां तैयार न हो जाये। इसके पश्चात वह डाटा अंतरण निर्देश देता है।

यह योजना काम कर रही है या नहीं इसके लिये युक्तियों (device) से एक संकेत उत्पन्न किया जाता है जिसका परीक्षण माइक्रोप्रोसेसर के द्वारा किया जाता है जो यह सुनिश्चित करता है कि युक्तियाँ तैयार है अथवा नहीं।

यह अन्तरण निम्न पदों में सम्पादित होता है

युक्तियो का तैयार रहने की प्रार्थना

युक्तियाँ तैयार है अथवा नहीं इस पर निर्णय करना

यदि तैयार हैं तो डाटा अन्तरण निर्देश देना और यदि नहीं तो वापस जाना युक्तियों के तैयार होने की स्थिति देखना

3. रूकावट के द्वारा चलाया गया अन्तरण (Interrupt driven transfer)

इस योजना का उपयोग उन 10 युक्तियों के साथ किया जाता है जो इस संबंध में काफी धीमी हो उदाहरण के लिये प्रिंटर या A/D परिवर्तक (Analog and Digital Convertor)

उदाहरण के लिये यदि हमें एक एनालॉग (Analog) वोल्टेज को डिजिटल वोल्टेज में परिवर्तित करना है तो सर्वप्रथम 10 युक्तियाँ आरंभ होती है माइक्रोप्रोसेसर अपन मुख्य प्रोग्राम के निर्देशों को सम्पादित करता है। इस दौरान प्रोसेसर बलपूर्वक एक प्रधान इनपुट लाइन में डाला जाता है जिसे रूकावट लाइन (Interrupt line) कहते हैं यह लाइन अपनी उच्च अवस्था में होती है। इससे माइक्रोप्रोसेसर अपने सामान्य प्रचालन से हट जाता है और एक विशिष्ट मेमोरी जगह में चला जाता है और जो अगला निर्देश सम्पादित होता है वह प्रोग्राम काउन्टर (Program Counter) में सुरक्षित हो जाता है।

8085 में पाँच रूकावट लाइन (Interrupt line) होती है। ये क्रमशः 7. 5,6.5, 5.5 तथा TRAP हैं। इनमें से कोई भी रूकावट बुलाई जाती है माइक्रोप्रोसेसर उस सबरूटीन (Subroutine) में कूद (Jump) जाता है जो एक स्थिर मेमोरी जगहों में होता है। यदि सिर्फ रूकावट मांगी जाती है तो किसी भी मेमोरी जगहों में चला जाता है।

5.6 सीधे मेमोरी अधिक देरी अन्तरण(Direct Memory Access transfer)

इससे पहले हमने देखा कि प्रोग्राम I/O तथा रूकावट I/O (Interrupt VO) डाटा अन्तरण (Transfer) में डाटा का अन्तरण माइक्रोप्रोसेसर से 10 युक्तियों के बीच होता है लेकिन कई ऐसे समय आते हैं जब डाटा का अन्तरण सीधे मेमोरी से वाहक युक्तियों के मध्य होता है और बहुत बडी मात्रा में डाटा का अन्तरण होता है । इस तरह का डाटा अन्तरण जो सीधे मेमोरी और I/O युक्तियों के मध्य हो सीधा मेमोरी मार्ग डाटा अन्तरण (Direct Memory Access transfer) कहलाता है। इस विधि का उपयोग करने से डाटा अन्तरण में काफी कम समय लगता है। (लगभग 1 माइक्रो सेकंड)

क्योंकि DMA, डाटा का अन्तरण मेमोरी से वाहक युक्तियों के बीच करती है अतः यह एक चिप होती है जो माइक्रोप्रोसेसर की तरह READ तथा WRITE प्रचालन को करती है।

एक निम्न पदों पर कार्य करता है :-

इनपुट युक्ति को निर्देशित करना

यह सुनिश्चित करना कि इनपुट युक्ति तैयार है कि नहीं यदि युक्ति तैयार नहीं है तो HALT अवस्था में जाना तथा वहाँ तब तक रहना जब तक कार्य करने वाला उसे START करे।

यदि इनपुट युक्ति तैयार है तो यह बताना कि युक्ति व्यस्त है अगर यह व्यस्त है तो तब तक चक्र में जाना जब वह मुक्त न हो जायें

यदि इनपुट युक्ति व्यस्त नहीं है तो एक READ आदेश युक्ति का जिसमें मेमोरी एड्रेस भेजना शुरू करें तथा साथ ही कितनी बाइट संख्या अन्तरण होना है यह भी बतायें।

इसके बाद प्रोग्राम के अन्य निर्देशों को सम्पादित करें।

DMA चिप माइक्रोप्रोसेसर को HOLD अवस्था में रखती है इस माइक्रोप्रोसेसर प्रोग्राम को सम्पादित करना बंद कर देता है और एड्रेस डाटा मेमोरी नियंत्रक लाइने उच्च अवरोध की अवस्था (High impedance state) में जाती है।

जैसे ही DMA चिप पर DMA स्वीकार होता है माइक्रोप्रोसेसर दूर तब RAM एड्रेस बस पर चला जाता है तथा नियंत्रक संकेत, नियंत्रक बस पर जाते है और इस तरह RAM तथा I/O युक्ति के मध्य डाटा अन्तरण होता है। तत्पश्चात् माइक्रोप्रोसेसर से HOLD अवस्था हटा ली जाती है।

माइक्रोप्रोसेसर के साथ दो प्रकार के DMA उपयोग में लाये जाते है जो निम्न है :-

(i) HALT DMA

(ii) इन्टरलीव DMA (Interleave DMA)

(i) HALT DMA

जैसा कि हम जानते हैं कि मेमोरी से I/O युक्तियों में डाटा के अन्तरण के समय माइक्रोप्रोसेसर HALT अवस्था में रहता है जब तक अन्तरण पूरी तरह पूरा नही हो जाता है। इस तरह के डाटा अन्तरण में दो प्रकार की विधियों इस्तेमाल में लाई जाती हैं।

(a) ब्लॉक अन्तरम DMA (Block Transfer DMA)

(b) चक्र चोरी कर्म DMA (Cycle Stealing DMA)

(a) ब्लॉक अन्तरण DMA (Block Transfer DMA)

इस तरह के अन्तरण में घेरे की युक्तियों (Peripheral device) DMA के लिये उसकी REQUEST LINE पर प्रार्थना (Request) करती है जो सीधे DMA चिप द्वारा माइक्रोप्रोसेसर से जुड जाता है। इसके पश्चात् माइक्रोप्रोसेसर वर्तमान निर्देश को सम्पादित करने के बाद HOLD संकेत भेजता है तथा DMA इसके बदले में DMA स्वीकार संकेत भेजती है। इस तरह DMA अन्तरण सम्पादित होता है।

(b) चक्र चोरी कर्म DMA (Cycle Stealing DMA)

इसमें DMA एक बाइट डाटा को मेमोरी तथा घेरे की युक्तियों (Peripheral device) के बीच अन्तरित करता है। इसमें DMA माइक्रोप्रोसेसर की एक क्लॉक चक्र को चुरा लेता है इसीलिये इसे चक्र चोरी क्रम DMA (Cycle Stealing DMA) कहते हैं। जैसा कि हम जानते हैं कि माइक्रोप्रोसेसर वाहन क्लॉक पल्स से चलाया जाता है तो इसे कुछ समय के लिये आसानी से रोका जा सकता है जब तक कि माइक्रोप्रोसेसर को क्लॉक पल्स न दी जाये।

(ii) इन्टरलीव DMA (Inter leave DMA)

यह काफी कठिन विधि है क्योंकि इसमें नियंत्रक माइक्रोप्रोसेसर की तंत्र बस (System bus) का इस्तेमाल करता है। इसका उपयोग उसी समय करता है जबकि माइक्रोप्रोसेसर तंत्र बस का इस्तेमाल न कर रहा हो माइक्रोप्रोसेसर जब आन्तरिक प्रचालन में व्यस्त रहता है तो वह अपनी तंत्र बस (System bus) का इस्तेमाल नहीं करता है और यही वह समय होता है जब DMA नियंत्रक इसका इस्तेमाल करता है।

5.7 अपवत्य सीधा मेमोरी मार्ग युक्ति (Multiple DMA device)

अभी तक हमने देखा कि सिर्फ एक युक्ति ही माइक्रोप्रोसेसर के साथ अन्तरमुख (Interface) होती है परन्तु देखा यह जाता है कि ऐसी कई युक्तियाँ माइक्रोप्रोसेसर अन्तरमुख होती हैं और इनमें से सिर्फ एक की प्रार्थना स्वीकार की जाती है। इसलिये यहाँ एक पीरियोरिटी (Priority) हल करने की यंत्र रचना (Mechanism) होती है जो केवल एक प्रार्थना को माइक्रोप्रोसेसर तक पहुंचाती है। सभी माइक्रोप्रोसेसर में यह सुविधा नहीं होती है। परन्तु अब यह विधि प्रभावपूर्ण रूप से अपनाई जाती है।

5.8 8085 तंत्र में पद DMA डाटा अन्तरण (DMA data transfer पद 8085 based System)

8085 चिप मे दो लाइन होती है जो DMA प्रचालन करती है।

a. HOLD LINE

b. HLDA LINE

जैसा कि हम देख चुके हैं कि HOLD लाइन DMA प्रार्थना के लिये उपयोग में लाई जाती है। और HLDA लाइन इस प्रार्थना की स्वीकारोक्ति होती है।

5.9 क्रमानुसार डाटा अन्तरण (Serially data transfer)

अभी तक हमने देखा कि युक्तियों में डाटा का अन्तरण एक बाइट या शब्द के द्वारा होता है, और यह अन्तरण समानान्तर क्रम में होता है। यदि कई क् जुडी हो और सभी समानान्तर क्रम में जुडी हो तो इस तरह के अन्तरण में का कार्य कठनाई होती है। यह असुविधाजनक होने के साथ-साथ काफी खर्चीला भी हो जा है। इसको दूर करने के लिये क्रमानुसार या एक के बाद एक डाटा भेजने का क कियाजाता है। इतनें डाटा एक बिट के बाद एक बिट भेजी जाती है। इससे का जमाव काफी कम होता है तथा लाइन चालक तथा प्राप्तकर्ता यदि उपयोग में ल गये है तो उनमें कमी आती है।

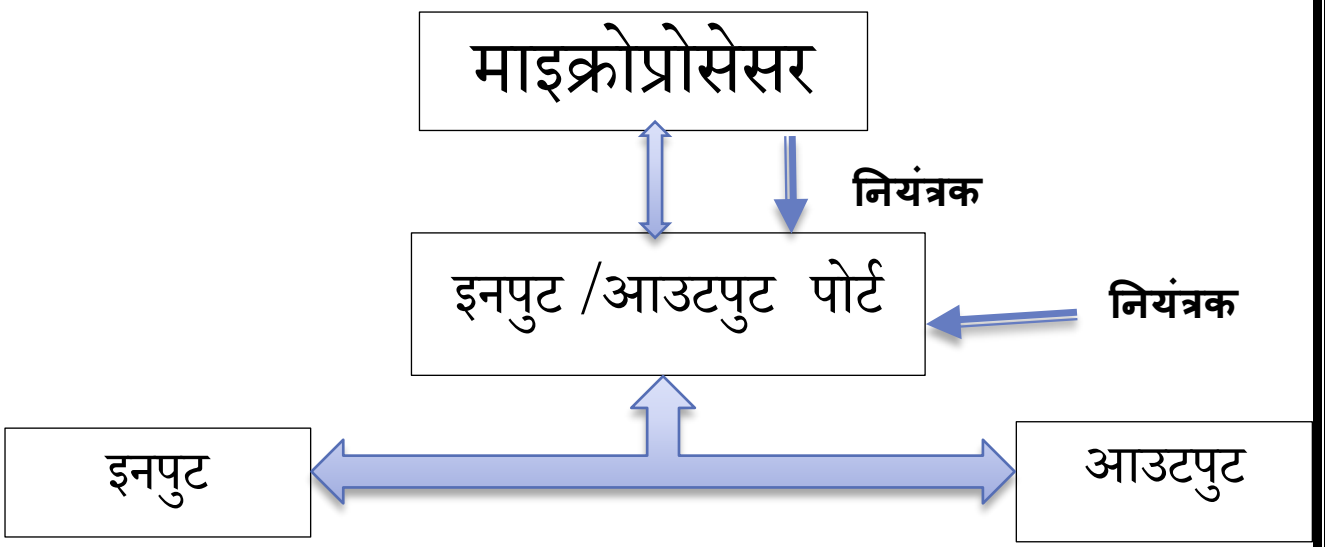
8085, SID (क्रमानुसार इनपुट डाटा) तथा SOD (क्रमानुसार आऊट डाटा) को स्वीकार करता है। एक निश्चित डाटा अन्तरण एक सॉटवेयर के माध्यम होता है जिसमें RIM तथा SIM निर्देश उपयोग में लाये जाते हैं। यह दोनो ही एक बाइट निर्देश है और जैसा कि हम देख चुके है कि इनका उपयोग READ MASK को सेट तथा रीसेट करने के लिये किया जाता है।

SIM निर्देश डाटा को SID में बलपूर्वक ले जाता है और इसमें ACC की MSB, 1 होती है। SIM निर्देश, SOD लाइन के लिये होती है जब डाटा बाहर जाता है तो इस समय ACC की MSB तथा 6 बिट 1 पर सेट होती है। इस तरह डाटा का क्रमानुसार अन्तरण होता है।

6. 8255 कोड और 8259 DMA IC

6.1 इनपुट / आऊटपुट पोर्ट (Input/Output Port)

यह एक ऐसी जगह होती है जहाँ से डाटा को चढाया (loading) तथा उतारा (Unloading) जाता है। दूसरे शब्दों में माइक्रोप्रोसेसर से डाटा को घेरे की युक्तिया (Peripheral) में जाना था यहाँ से माइक्रोप्रोसेसर में आना। पोर्ट एक ऐसी जगह होती है जहाँ माइक्रोप्रोसेसर तथा घेरे की युक्ति (Peripheral) से आये डाटा को अस्थाई रूप से रखा जाता है और फिर यहाँ से वाहन युक्तियों तथा माइक्रोप्रोसेसर में भेजा जाता है। इसे चित्र (6.1) द्वारा आसानी से समझाया जा सकता है। एक I/O पोर्ट मे डाटा अन्तरण की कई लाइने होती हैं तथा इनमें कुछ नियंत्रक लाइन भी होती हैं।



चित्र (6.1) इनपुट / आऊटपुट पोर्ट

6.2 प्रोग्राम सहित I/O पोर्ट (Programmable I/O Port)

इस व्यवस्था में पोर्ट का इनपुट / आऊटपुट निर्देशों के द्वारा प्रोग्राम किया जाता है। यह निर्देश IN OUT होते हैं।

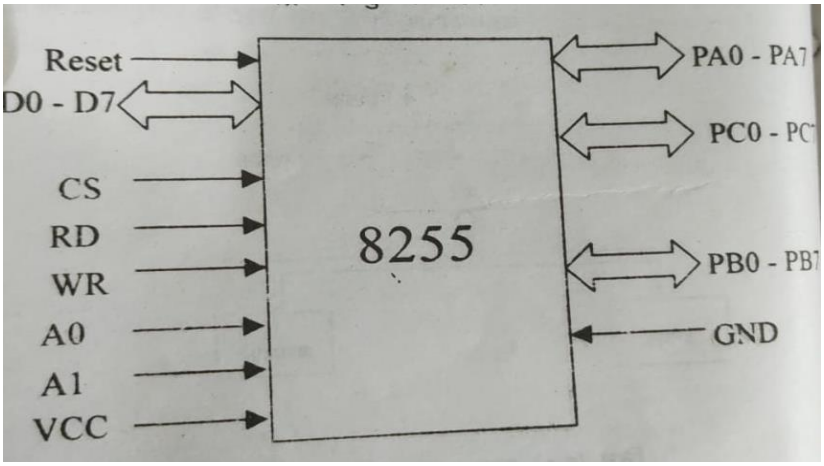
सामान्यतः पार्ट को निम्न एक या एक से अधिक गुणा के आधार पर प्रोग्राम किया जा सकता है :

1. सरल इनपुट और आऊटपुट विधि
2. स्ट्रोब इनपुट और आऊटपुट विधि
3. नियंत्रण विधि

सरल इनपुट और आऊटपुट विधि में इनपुट को इनपुट युक्ति तथा आऊटपुट को आऊटपुट युक्ति के द्वारा लेच (Latch) कर दिया जाता है। स्टेश इनपुट और आऊटपुट विधि में डाटा को पोर्ट पर लाने और ले जाने के लिए वाहक स्ट्रोब सिग्नल की आवश्यकता होती है। नियंत्रण विधि में अलग पिनों के द्वार दूसरे पोर्ट में डाटा अन्तरण को नियंत्रित किया जाता है।

6.3 प्रोग्राम के घेरे में अन्तरमुख: 8255 (Programmable Peripheral Interface: 8255)

8255 पोर्ट चित्र को चित्र (6.2) में दर्शाया गया है। इसमे तीन के A, B, C होते हैं जिसमे एड्रेस इनपुट होते है।



चित्र (6.2) 8255 पोर्ट

माइक्रोप्रोसेसर से 8255 में और 8255 से माइक्रोप्रोसेसर में डाटा का अन्तरण D0-D7 डाटा लाइनों के माध्यम से होता है। 8255 को प्रोग्राम करने लिए इसमें उपयोग में आने वाले पोर्ट को निम्न समूह मे उपयोग में लाते है।

समूह A- पोर्ट A तथा पोर्ट C की अधिकतम अर्थपूर्ण (Most Significant) चार बिट

समूह B- पोर्ट B तथा पोर्ट C की न्यूनतम अर्थपूर्ण (Least Significant) चार बिट

इस चिप में A0 तथा A1 दो इनपुट संकेत उपयोग में लाये जाते हैं जो इस बात का निर्धारण करते हैं कि किस पोर्ट से डाटा का अन्तरण होना है। यदि A0 = 0, A1 = 0 हो तो पोर्ट A से डाटा पढ़ा जाता है तथा इसी पोर्ट में डाटा को लिखा भी जा सकता है। यदि A0 = 0, A1 = 1 है पोर्ट B में पढ़ा और लिखा जा सकता है। यदि A0 = 1, A1 = 0 हो तो पोर्ट C पर डाटा पढ़ा और लिखा जा सकता है। पोर्ट चिप 8255 में कार्य करने की तीन प्रचालित विधियाँ हैं :

(i) विधि शून्य (Mode 0)

(ii) विधि एक (Mode 1)

(iii) विधि दो (Mode 2)

विधि शून्य (Mode 0)

यह एक सरल इनपुट / आऊटपुट विधि है इसमें पोर्ट A, B, C के ऊपरी भागों तथा पोर्ट C से निचले भाग का इस्तेमाल किया जाता है। दूसरे शब्दों में इसमें पोर्ट A, B. C की ऊपरी चार अधिकतम अर्थपूर्ण बिट (Most Significant) तथा पोर्ट C की चार न्यूनतम अर्थपूर्ण (Least Significant) बिट का उपयोग किया जाता है।

विधि 0 (Mode 0) में डाटा के पढ़ने और लिखने में किसी बाह्य नियंत्रण संकेत (Control Signal) की आवश्यकता नहीं होती है। इस विधि का इस्तेमाल करने में महत्वपूर्ण बात यह है कि जब एक पोर्ट, आऊटपुट पोर्ट की तरह काम करता है तो जो डाटा इस पोर्ट में लिखा जाना है वह लेच (Latch) हो जाता है और यदि यह पोर्ट इनपुट की तरह काम करता है तो जो डाटा लिखा जाना है यह लैच (Latch) न होकर बफर (Buffer) में चला जाता है।

विधि एक (Mode 1)

इसे स्ट्रोब इनपुट / आऊटपुट मोड भी कहते है। इस विधि में केवल पोर्ट A तथा B को ही प्रोग्राम किया जाता है। इसके लिये कुछ नियंत्रण संकेतो की आवश्यकता पड़ती है ।

यदि पोर्ट A इनपुट की तरह इस्तेमाल होता है तो पोर्ट C की तीसरी से पांचवी पिन (PC3-PC5) नियंत्रण संकेतों की तरह कार्य करती है। यदि पोर्ट B इनपुट की तरह इस्तेमाल होता है तो पोर्ट C की शून्य से दूसरी पिन (PC0-PC2) नियंत्रण संकेतों की तरह कार्य करती है तथा छठवी पिन का उपयोग नहीं किया जाता है।

जब पोर्ट A का इस्तेमाल आऊटपुट की तरह किया जाता है तो C की तीसरी छटवीं तथा सातवी पिने नियंत्रण संकेत का कार्य करती है। पार्ट B के लिये शून्य पहली तथा दूसरी पिनें नियंत्रण संकेत का कार्य करती

विधि दो (Mode 2)

इस विधि में सिर्फ पोर्ट A का इस्तेमाल किया जाता है तथा के तीसरी पिन से लेकर सातवीं पिन (PC3-PC7) नियंत्रण संकेतो का कार्य करत है। इस विधि में पोर्ट A किसी घेरे की युक्ति (Peripheral device) के लिये भेजा जाता है तथा इसे ग्रहण भी करता है। इसमें स्ट्रोब संकेत (STB) होता है जो एक द्वारा डाटा को स्ट्रोब किया जाता है।

सेट / रीसेट लक्षण (Set / Reset Feature)

इस लक्षण की मदद से माइक्रोप्रोसेसर की सहायता से पोर्ट C की किसी भी बिट को नियंत्रित किया जा सकता है। दूसरे शब्दों में पोर्ट C की 8 बिट मे से किसी एक बिट को सेट तथा रीसेट किया जा सकता है। यह 8255 को आऊटपुट पिन होती है।

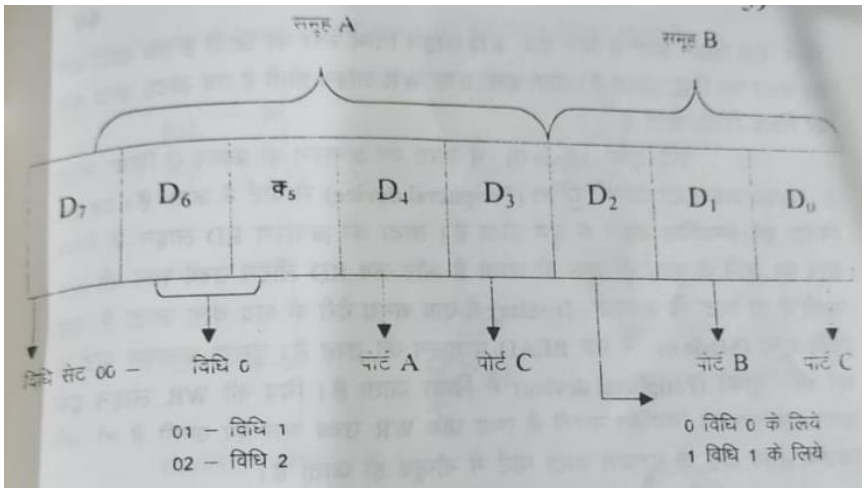
उदाहरण के लिये यदि 8 बिट मे 8 लाइट उत्सर्जक डायोड लगे हो तो चयनानुसार उन्हें ON या OFF किया जा सकता है।

6.4 8255 को प्रोग्राम करना (Programming the 8255)

माइक्रोप्रोसेसर के द्वारा एक नियंत्रक बाइट को लिखकर 8255 का प्रोग्राम किया जाता है। जब 8255 के A0 तथा A1 संकेत उच्च स्तर पर होते हैं। यह 8255 के लिखने प्रचालन के लिए होता है और इससे 8255 के नियंत्रक जिस्टर को निर्देशित करते हैं।

इस तरह 8255 को रीसेट (Reset) किया जा सकता है जब इस पर चिप रीसेट (Reset) लाइन उच्च स्तर पर हो और यह पॉवर चालू होने से लगभग $50\mu s$ तक हो।

इस समय सभी पोर्ट विधि 0 पर सेट होते हैं। चित्र (6.3) में नियंत्रक का फार्मा (Format) दर्शाया गया है।



चित्र (6.3) विधि नियंत्रक शब्द का खाका

चित्रानुसार निम्न अर्थपूर्ण (Least Significant) तीन बिट समूह B पर आधारित पोर्ट की विधि को बताते हैं। बिट D3 -D6 समूह A के पोर्ट की विधि को बताते है। D7 बिट 1 या उच्च स्तर पर होनी चाहिये जो 8255 को यह दर्शाती है कि ये विधि नियंत्रक शब्द है। इस तरह I/O की एड्रेस को देने के लिए I/O मैप्ड I/O योजना का इस्तेमाल किया जाता है।

6.5 8255 के समय प्रचालन (Timing Operation of 8255)

8255 की मदद से चार तरह के डाटा अन्तरण सम्भव होते हैं और 8255 के प्रचालन में इन्हीं के समय की आवश्यकता होती है।

1. माइक्रोप्रोसेसर से 8255 में पढ़ने तथा लिखने के लिये प्रचालन

2. विधि शून्य (Mode 0) अन्तरण

3. विधि शून्य (Mode 0) अन्तरण

4. विधि दो (Mode 2) अन्तरण

माइक्रोप्रोसेसर तथा 8255 के मध्य डाटा का अन्तरण होने के लिये READ तथा WRITE दो प्रचालन तथा उनके समय पर ध्यान दिया जाता हैं। READ प्रचालन के लिये आवश्यक है कि सर्वप्रथम चिप इस प्रचालन के लिए तैयार हो जिसके लिये CS लाइन होती है और जब RD लाइन निम्न स्तर पर जाती है तब डाटा पर डाटा पढ लिया जाता है। ठीक इसी तरह WR लाइन होती है तब डाटा, डाटा पर लिख दिया जाता है।

विधि शून्य (Mode 0) में डाटा का अन्तरण दो प्रकार से किया जाता है। पहला जब डाटा घेरे की युक्ति (Peripheral device) से पोर्ट में जाता है। यह IN निर्देश को सम्पादित करने से पूर्ण होता है। डाटा का अन्तरण RD लाइन के निम्न स्तर पर जाने के साथ ही शुरू हो जाता है और जब RD लाइन उच्च स्तर की ओर जाती है तो डाटा त्रि अवस्था (Tristate) में एक समय देरी के बाद चला जाता है। पर विधि शून्य (Mode 0) में एक READ प्रचालन की तरह है। दूसरा अन्तरण पोर्ट से घेरे की युक्ति (Peripheral device) में किया जाता है। चिप की WR लाइन इस डाटा अन्तरण को नियंत्रित करती है तथा जब WR उच्च स्तर पर जाती है तो अधिकतम समय देरी के पश्चात डाटा पोर्ट में मौजूद हो जाता है।

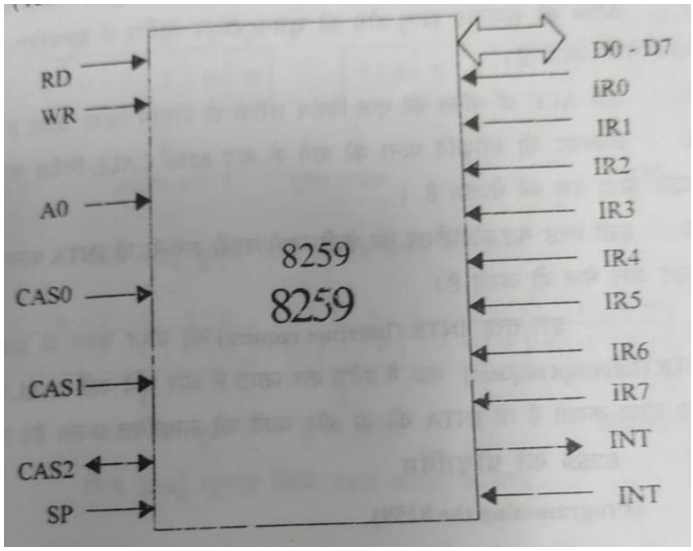
विधि 1 (Mode 1) के अनुसार भी डाटा का अन्तरण दो प्रकारों से किया जाता है। जैसा कि हम जानते हैं कि इस विधि में स्ट्रोब लाइन का इस्तेमाल किया जाता है। जब पोर्ट A के लिये C पोर्ट की चौथी बिट निम्न स्तर पर होती है। तब इनपुट पर स्ट्रोब संकेत STB होता है तथा डाटा का अन्तरण होता है। यहाँ होता । यह है कि जब इनपुट युक्ति

स्ट्रोब पर क्रिया करती है तो बाह्य परिपथ को निर्देशित करती है जो पोर्ट A के लिये C पोर्ट की पांचवी बिट जो इनपुट बफर फुल (IBF. Input buffer full) को सेट किये जाने से होती है। अर्थात इसका स्तर उच्च होता है। इसमें माइक्रोप्रोसेसर की INTR (Interrupt) लाइन का उपयोग होता है और जब यह लाइन उच्च स्तर पर जाती है तो डाटा पढ (Read) लिया जाता है। आऊटपुट डाटा अन्तरण में माइक्रोप्रोसेसर एक बाइट को लिखता है और जब डाटा लिख लिया जाता है तो WR लाइन उच्च स्तर पर होती है OBF (Output buffer full) निम्न स्तर पर चला जाता है।

विधि 2 (Mode 2) में पोर्ट A द्वि दिग्दर्शक बस को बताती है, जब यह 2 के लिये प्रोग्राम की जाती है। इस विधि का इस्तेमाल करके डाटा का अन्त माइक्रोप्रोसेसर से I/O युक्तियों में तथा I/O युक्तियों से माइक्रोप्रोसेसर में दोनो तरफ किया जाता है।

6.6 प्रोग्राम रूकावट नियंत्रक 8259 (Programmable Interrupt Controller: 8259)

जब रूकावट द्वारा चलाये गये डाटा अन्तरण का प्रयोग करके I/O युक्ति माइक्रोप्रोसेसर के साथ अन्तरमुख होती है तो चुनाव योजना या वेक्टर रूकावट योजना(Vectored Interrupt Scheme) का इस्तेमाल किया जाता है।



चित्र (6.4) प्रोग्राम रुकावट नियंत्रक

जैसा कि चित्र (6.4) में दर्शाया गया है। 8259 के प्रचालन निम्न पदों में पूर्ण होता है :

1. 8259, I/O युक्ति से एक रूकावट संकेत (Interrupt) प्राप्त करता है तथा इसकी पिरीयोरिटी (Periority) तथा मास्क (Mask) को परीक्षण करने के पश्चात माइक्रोप्रोसेसर को रोक (Interrupt) देता है ।

2. रूकावट (Interrupt) को प्राप्त करने के बाद माइक्रोप्रोसेसर प्रोग्राम रूकावट नियंत्रक के माध्यम से इसे स्वीकार करता है।

3. स्वीकृति संकेत के पश्चात, प्रोग्राम रूकावट नियंत्रक तीन बाइट का CALL निर्देश भेजता है।

4. इस समय IR इनपुट 100 us के ऋणात्मक पल्स द्वारा चलाये जा सकते है।

5. रुकावट की स्वीकृति संकेत पाने के पश्चात इससे संबंधित ISR (Interrupt Subroutine) बिट अपनी निम्नतम पिरीयोरिटी पर सेट होती है।

6. (Interrupt Service Subroutine) रूकावट आदेश के अंत होने की सूचना भेजती है।

7. 8259 को रूकावट खत्म होने की सूचना OUT निर्देश के सम्पादित होने साथ भेजी जाती है।

8. यहाँ ACC के पंक्ति को एक विशेष तरीके से प्रोग्राम किया जाता है।

9. रूकावट की स्वीकृति पल्स को पाने के बाद 8259 CALL निर्देश की पहली बाइट डाटा बस को भेजता है ।

10. इसी तरह माइक्रोप्रोसेसर को भेजी जाने वाली अगली दो INTA पल्स की ट बाइट और भेज दी जाती है।

इस तरह INTR (Interrupt request) को प्राप्त करने के बाद 80 INTR (Interrupt request) चक्र में प्रवेश कर जाता है और यदि वहाँ CALL प्रचालन कोड प्राप्त करता है तो INTA की दो और चक्रों को सम्पादित करता है।

6.7 8259 की प्रोग्रामिंग (Programming the 8259)

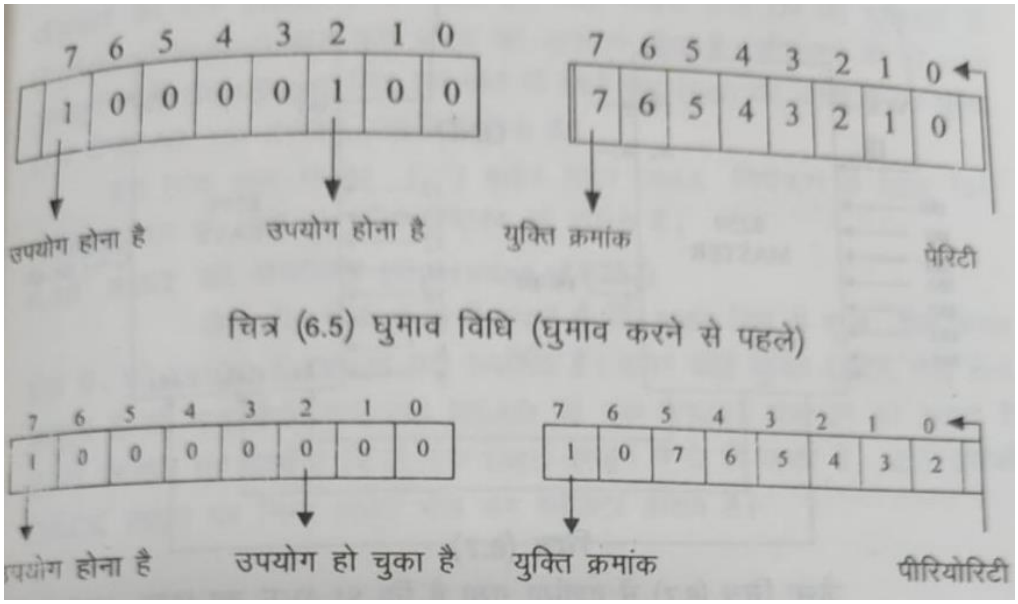
8259 को सर्वप्रथम शुरू करने के लिये ICW (Initialization Command Word) का इस्तेमाल किया जाता है। इसकी प्राथमिकता (Initialization) के समय ISS (Interrupt Service Subroutine) के एड्रेस को प्रत्येक युक्ति के लिये पूर्व में ही ले लिया जाता है। ये एड्रेस एक दूसरे से 4 से 8 बाइट की दूरी पर हो सकते हैं। और 64 K के मेमोरी जगह इस्तेमाल में लाई जा सकती है।

8259 के A0 इनपुट का उपयोग RD तथा WR इनपुट के साथ प्रोग्रामिंग के लिये किया जाता है। जैसे ही 8259 की प्राथमिकता सेट होती हैं। इसके बाद 8259 Fully

nested Periority में चला जाता है जहाँ सारे इनपुट पेरिटी क्रम 0 से 7 में सेंट हो जाते है। दूसरी पेरिटी भी सेट की जाती है परन्तु उसके लिये OCW (Other Command Word) भेजे जाते हैं।

दूसरी विधि में 8259 को प्रोग्राम किया जाये तो यह विधि घुमावदार पिरियोरिटी कहलाती है। यह दो प्रकार की होती है :

1. **स्वतः घुमाव (Auto Rotate)**
2. **विशिष्ट घुमाव(Specific Rotate)**

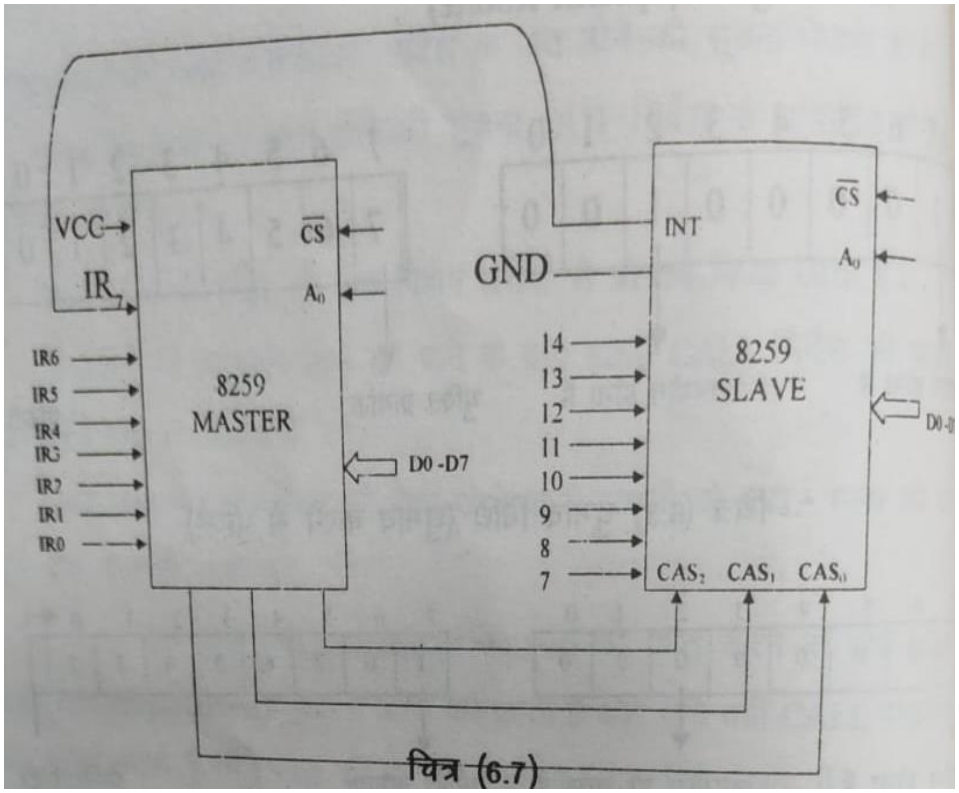


चित्र (6.6) घुमाव विधि EOI आज्ञा के बाद

जैसा कि चित्र (6.6) में दर्शाया गया है कि घुमाव करने से पहले एक शब्द तथा उससे संबंधित पीरियोरिटी (Periority) चित्रानुसार सेट है पीरियोरिटी 7 पर सेट है तथा शब्द मे 2 तथा 7 बिट का उपयोग होना है। EOI आज्ञा के बाद 2 विट का उपयोग हो चुका होता है और यह निम्न स्तर पर चली जाती है तथा पीरियोरिटी (Periority) की निम्न अवस्था 7 पर आ जाती है जो युक्ति 1 के लिये कार्य करती है। कोई अन्य युक्ति कार्य नही करती है जब तक आज्ञा सम्पादित नहीं होती है ।

6.8 कई 8259 को जोडना (Cascading of 8259s)

लगभग आठ 8259 एक दूसरे के साथ जोड़ा जा सकता है जो एक Master के साथ Slave इकाई की तरह कार्य करते हैं और इस तरह 64 I/O युक्तियों 8259 की रुकावट चालक विधि द्वारा माइक्रोप्रोसेसर से जोड़े जा सकते है। चित्र (6.7) मे प्रदर्शित है दो 8259 चिप जिनमें से एक Master तथा दूसरा Slave की तरह कार्य करता है । इस तरह 15 I/0 युक्तियों 8085 के साथ अन्तरमुख (Interface) होती है।



चित्र (6.7)

जैसा चित्र (6.7) में दर्शाया गया है कि SLAVE का INT, आऊट IR, इनपुट के साथ जुडा हुआ है। अब बची हुई सात विधि (IR0 - IR6) होती है क्रिया का क्रम MASTER चिप द्वारा लिया जायेगा ।

यदि 7 से 14 दिन SLAVE के इनपुट interrupt होते है तो SLAVE द्वारा रुकावट (Interrupt) माइक्रोप्रोसेसर को दी जाती है बशर्ते कोई उच्च Periority बची न हों। जब INTA संकेत प्राप्त होता है तो MASTER, CALL निर्देश की पहली बाइट भेजता है । इसके बाद SLAVE शक्तियुक्त होता है तथा एक निर्देश को उत्पन्न करता है जो CAS लाइनो पर दिये जाते है। और इस तरह SLAVE, CALL निर्देश की अगली दो बाइट सीधे माइक्रोप्रोसेसर को भेजता है उसकी अगली INTA पल्स के लिये।

6.9 प्रोग्राम किया हुआ DMA नियंत्रक 8257 (Programmable DMA Controller: 8257)

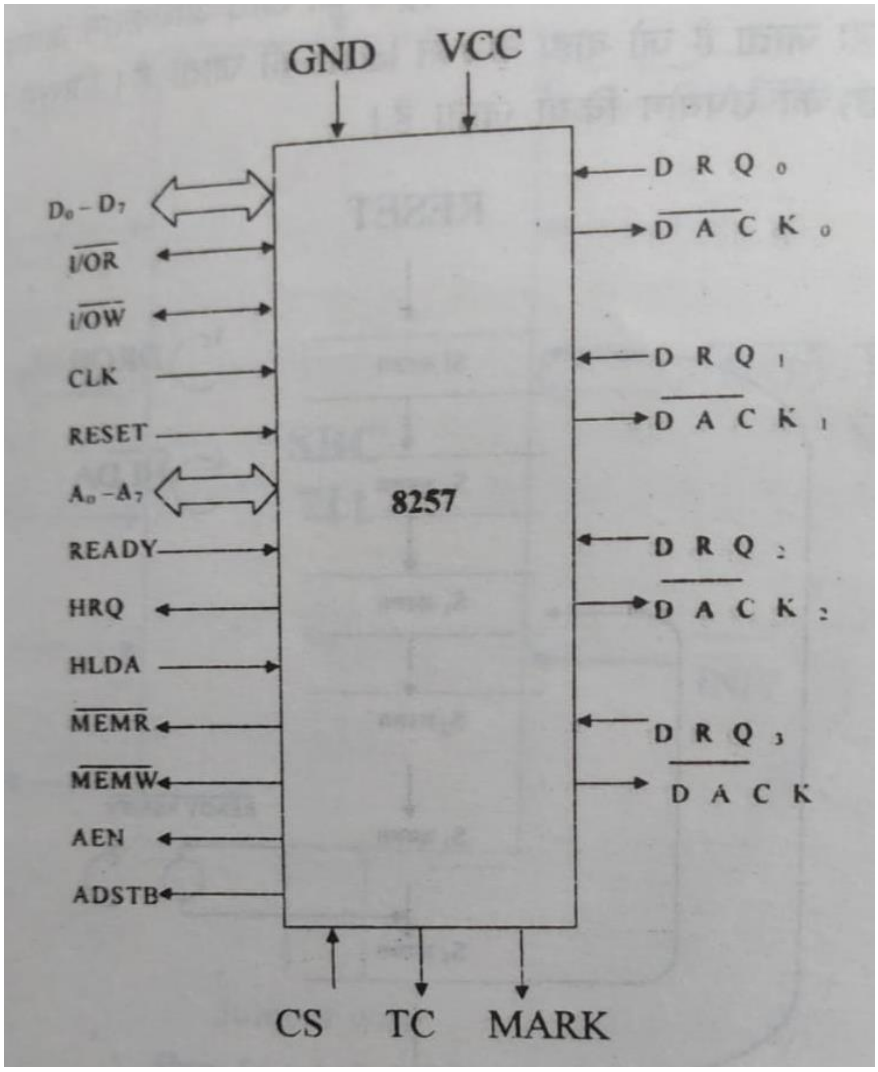
जैसा कि पूर्व में हमने देखा कि वे I/O युक्तियां जिनको DMA प्रार्थना (Request) दी जाती है ये बसों (Buses) पर अपना नियंत्रण प्राप्त करती है इसका मतलब यह हुआ कि I/O युक्तियों के अपने खुद के रजिस्टर्स होते हैं मेमोरी निर्देशों को रखते हैं और इसके साथ ही डाटा अन्तरण के दौरान बाइट गिनते है।

3257 को इस तरह बनाया जाता है कि ये वार I/O युक्तियों के साथ अन्तरमुख (Interface) हो सके 8257 की अपनी चार चैनल होती हैं और प्रत्येक चैनल के 16 बिट रजिस्टर्स होते है इन 16 बिट रजिस्टर्स में से 14 बिट दूसरे रजिस्टर को रोक (hold) करके रखते है जहाँ मेमोरी तथा घेरे की युक्तियों के (Peripheral devices) मध्य कुल बाइट का अन्तरण होना है। रजिस्टर की दो उच्च अर्थपूर्ण(Most Significant) बिट इस बात के लिये सेट (Set) की जाती है कि युक्ति द्वारा चैनल पर क्या प्रचलन प्रचालित होना है।

इस तरह कुल 16384 (2^{14}) बाइट डाटा DMA नियंत्रण के तहत बिना माइक्रोप्रोसेसर के आये हुय सीधे अन्तरण हो सकते है।

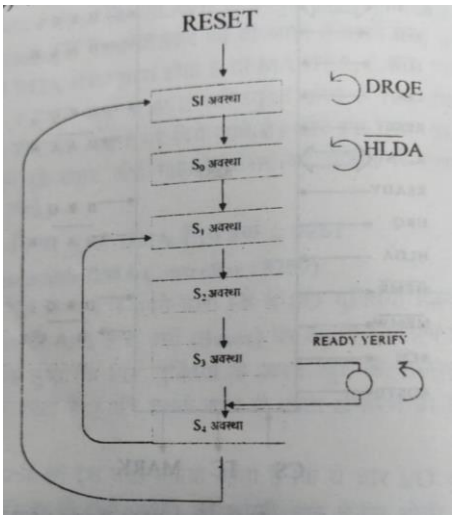
6.10 8257 का प्रचालन (Operation of 8257)

जैसा कि चित्र (6.8) में स्पष्ट है कि 8257 चिप में कुल चार चेनल होते हैं जो DRQ0 से DRQ3 तक प्रदर्शित है। अगर कोई युक्ति DMA सेवा लेना चाहती है तो माइक्रोप्रोसेसर एक READ या एक WRITE प्रचालन को रखता है इसके पश्चात् यह प्रार्थना किसी एक DRQ लाइन में दे दी जाती है। 8257 इसको DACK लाइन पर निम्न संकेत भेज कर स्वीकार करता है।



8257 का प्रचालन (Operation of 8257)

8257 के प्रचालन को एक अवस्था परिवर्तन (Transition) द्वारा समझा जा सकता हैं। चित्र (6.9) स्पष्ट है कि जब 8257 चिप एक DMA प्रार्थना सुनने बाद SI अवस्था (Idlestate) से S0 अवस्था में जाती है। S1 अवस्था वह अवस्थ होती है जब माइक्रोप्रोसेसर RST रूकावट को जानने के बाद बस इडल (BI) अवस्थ में प्रवेश करता है। S0 अवस्था में माइक्रोप्रोसेसर को HRQ लाइन के माध्यम HOLD प्रार्थना (request) भेजी जाती है और यह इसी अवस्था में रहती है जब इसका स्वीकार संकेत (acknowledgement) प्राप्त नहीं होता है स्वीकार संकेत हो जाने के बाद उच्चतम Periority DACK लाइन के माध्यम से भेजी जाती है इसके पश्चात् 8257 S0 अवस्था में प्रवेश करता है। S2, S3, S4 अवस्थायें डाटा अन्तरण को नियंत्रित करती है। यह अवस्था माइक्रोप्रोसेसर की T1, T2, T3 अवस्थ के समान होती है। जहाँ डाटा READ तथा WRITE किया जाता है। मेमोरी एड्रेस एड्रेस तथा डाटा बस पर भेजा जाता है। एड्रेस की आठ अधिकतम अर्थपूर्ण बिट डाटा बस में भेजा जाता है जो बाह्य रूप से latch की जाती है जिसमे एड्रेस संकेत (ADSTB) का उपयोग किया जाता है।

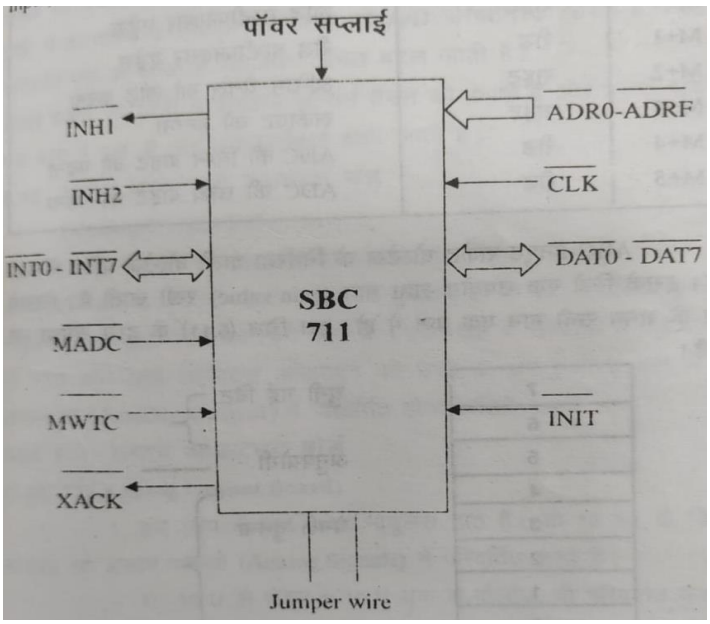


6.11 समान इनपुट सहायक तंत्र (Analog input Subsystem)

इस तरह के सहायक तंत्र को डाटा लाभ तंत्र कहते हैं। यहाँ माइक्रोप्रोसेसर के द्वारा एक समान वोल्टेज को पढ़ने की आवश्यकता होती है और फिर इसे समतुल्य डिजिटल रूप में बदलना पड़ता है।

समान इनपुट बोर्ड :(SBC 711) (Analog Input Board: SBC 711)

यह एक अकेले बोर्ड की युक्ति है और यह लगभग 16 अनित्य को बदलता है जिसमे लगभग 8 अनित्य (Variables) को घटते क्रम में (Variables) बदलता है। इसको 8085 माइक्रोप्रोसेसर के साथ जोड़ा जा सकता है जो मेमोरी मे पड़ विधि में अन्तरमुख होता है इनपुट चेनल पर आने वाले समान इनपुट (Analog Input) को 12 बिट के डिजिटल (Digital) रूप में बदल सकता है।



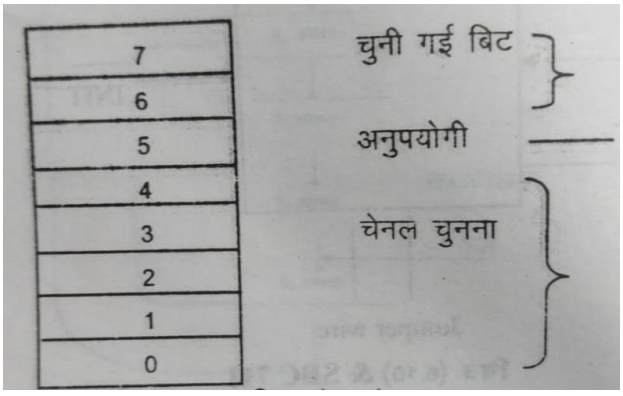
चित्र (6.10) के अनुसार समान संकेतो (Analog Signals) की Jumper तारे की से सीधा बोर्ड में जोड़ा जाता है। SBC 711 में 16 मेमोरी जगह ह M+5 के रूप में लिखा जाता है (Table- 1)A यह चिप एड्रेस की जाती है INHI तथा

INH2, RAM तथा ROM के लिये संकेत उत्पन्न करती है। समयकाल (Time duration) निश्चित होते हैं (लगभग 0.05 1s 1.5 ms) इस चिप एक अन्तरण स्वीकार संकेत (Transfer acknowledge Signal, XACK) होता है संकेतो की चौड़ाई को नियंत्रित करता है।

तालिका नंबर एक

एड्रेस	आदेश	फलन
M+0	राइट	लोड आदेश रजिस्टर
M+0	रीड	रीड अवस्था रजिस्टर
M+1	राइट	लोड मल्टीप्लेक्सर एड्रेस
M+1	रीड	रीड मल्टीप्लेक्सर एड्रेस
M+2	राइट	अंतिम चैनल को लोड करना
M+3	राइट	रुकावट को हटाना
M+4	रीड	ADC की निम्न बाइट को पढ़ना
M+5	रीड	ADC की उच्च बाइट को पढ़ना

ABC इनपुट संकेत वोल्टेज के निर्भरता वाली वोल्टेज पंक्ति में स रखी जाती है। इसके लिये एक उपयोग लाभ मान (gain value) रखी जाती है। जिस परिवर्तन के समय सभी मान एक क्रम में हो। इस चित्र (6.11) के द्वारा समझा सकता है।



जैसा कि चित्र (6.11) से स्पष्ट है कि एक शब्द मे प्रथम पाँच या निम्न क्रम की बिट चैनल का चुनाव करती है जो 1 से लेकर 32 तक कुछ भी हो सकती है। इसके पश्चात् 6 तथा 7 बिट लाभ (gain) का निर्धारण करती है। यदि 7 तथा 6 बिट के क्रम 00, 01, 10, 11 है तो इस पर लाभ (gain) क्रमश: 1, 2, 4, 8 होगें।

परिवर्तन के प्रचालन की तीन विधियों होती हैं-

1. एकाएक उत्पन्न चैनल इनपुट (Random Channel Input)
2. एक के बाद एक चैनल इनपुट (Single repetitive Channel Input)
3. एक निश्चित क्रम में चैनल इनपुट (Sequential Channel Input)

प्रथम प्रकार के एकाएक उत्पन्न चैनल इनपुट में प्रोसेसर किसी भी चैनल का प्रयोग परिवर्तन की लिये कर सकता है। एक के बाद एक चैनल इनपुट विधि में आन्तरिक क्लॉक पल्स होती है जो A/D परिवर्तन को करती है और जब परिवर्तन पूरा हो जाता है तब स्वतः चैनल बदल जाती है। तीसरे प्रकार की विधि मे प्रोग्रामर अन्तिम चैनल को करता है और इसके बाद चैनल एक क्रम में एक के बाद एक Select होती जाती है।

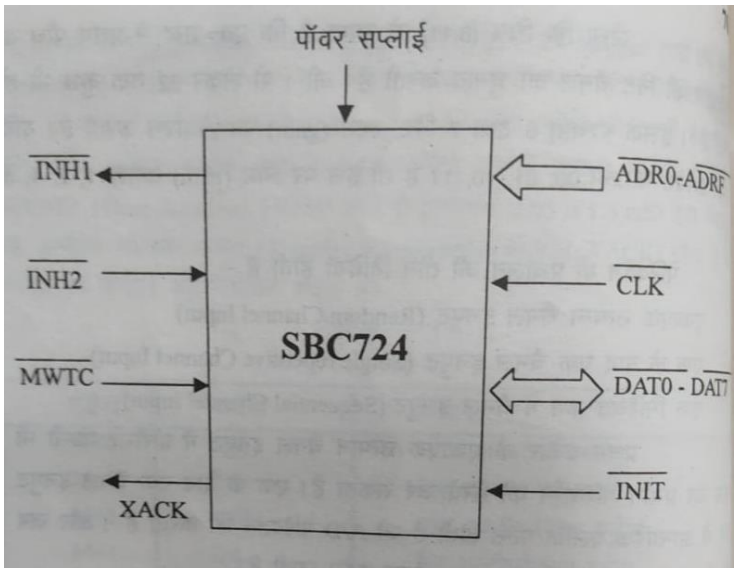
6.12 समान आऊटपुट सहायक यंत्र (Analog Output Subsystem)

जितने भी माइक्रोप्रोसेसर आधारित नियंत्रक तंत्र हैं। जिसमे जितने कार्य किये जाते है चाहे वो वोल्टेज नियंत्रण युक्तियां हो या रीले प्रचालित युक्तियां हो माइक्रोप्रोसेसर द्वारा नियंत्रित किये जाते है। इस तरह के उपयोगों में माइक्रोप्रोसेसर से प्राप्त होने वाले डिजिटल आऊटपुट को उसी के समानुपातिक क्रम में समान आऊटपुट (Anaolg output) में परिवर्तित होना चाहिये ।

SBC 724 समान आऊटपुट बोर्ड (SBC 724 Analog Output Board)

इस चिप में चार DAC माडुलस होते है जो 12 bit के डिजिटल सकेतो को समान संकेतो (Analog Signals) में परिवर्तित करते हैं।

ये -10V से लेकर + 10 V तक के वोल्टेज को परिवर्तित कर सकते हैं।



चित्र (6.12) – SBC 724 समान आऊटपुट बोर्ड

SBC 724 चिप चित्र (6.12) में प्रदर्शित है जिसमें ADRO-ADRF की 12 लाइनें प्रदर्शित है। जब माइक्रोप्रोसेसर चिप के साथ मेमोरी राइट आदेश के साथ जुड़ता है । इस बोर्ड से एड्रेस को जम्पर तारों की सहायता से जोड़ा जाता है। जो 8 के गुणक में होते हैं। जब माइक्रोप्रोसेसर से डाटा भेजा जाता है तो ये DAC द्वारा लेच (latch) कर दिया जाता है इसके पश्चात् परिवर्तन शुरू हो जाता है ये लेच तब शुरू होता है जब उच्च बाइट का अन्तरण किया जाता है। निम्न बाइट HOLD रजिस्टर्स में रखे जाते हैं जो सभी DAC के लिये एक जैसे होते हैं ।

आऊटपुट लिखावट (Output Writing)

प्रत्येक DAC के आऊटपुट को एक लोड प्रतिरोध के समानान्तर क्रम में लगाते हैं। इस प्रतिरोध का मान DAC से आने वाले आऊट वोल्टेज तथा धारा के द्वारा निकाला जाता है। DAC के बचाव के लिये शार्ट परिपथ का इस्तेमाल किया जाता है।

7. 8086 माइक्रोप्रोसेसर आर्किटेक्चर और ऑपरेशन

8086 माइक्रोप्रोसेसर आर्किटेक्चर और ऑपरेशन:

यह 16 बिट का माइक्रोप्रोसेसर है। 8086 में 20 बिट एड्रेस बस है जो 220 मेमोरी स्थानों तक पहुंच सकती है)I एमबी (| यह 64K इनपुट आउटपुट/ पोर्ट तक का समर्थन कर सकता है। यह 14, 16-बिट रजिस्टर प्रदान करता है। यह एड्रेस और डेटा बस एडी०-एडी -15 और ए-16 – ए-19 को मल्टीप्लेक्स करता है । इसके लिए 33% के साथ एकल चरण घड़ी की आवश्यकता होती जो आंतरिक समय प्रदान करने के लिए कर्तव्य चक्र है । 8086 को दो मोड में संचालित करने के लिए डिज़ाइन किया गया है, न्यूनतम और अधिकतम। यह मेमोरी से 6 निर्देश बाइट्स तक प्रीफ़ेच कर सकता है और उन्हें कतार में खड़ा कर सकता है निर्देश निष्पादन में तेजी लाने के लिए। इसके लिए +5 वोल्ट बिजली की आपूर्ति की आवश्यकता होती है। यह एक 40 पिन दोहरी लाइन पैकेज है ।

न्यूनतम और अधिकतम मोड:

न्यूनतम मोड MN / MX # इनपुट पिन पर तर्क I लागू करके चुना जाता है। यह एक है एकल माइक्रोप्रोसेसर कॉन्फ़िगरेशन। लॉजिक O को लागू करके अधिकतम मोड का चयन किया जाता है

8086 की आंतरिक आर्किटेक्चर

8086 में दो ब्लॉक बीआईयू और यूरोपीय संघ हैं। बीआईयू निर्देश जैसे सभी बस संचालन करता है स्मृति के लिए ऑपरेंड लाना, पढ़ना और लिखना और उसके पते की गणना करना मेमोरी ऑपरेंड। निर्देश बाइट्स को निर्देश कतार में स्थानांतरित कर दिया

जाता है। यूरोपीय संघ अनुदेश प्रणाली बाइट कतार से निर्देश निष्पादित करता है। दोनों इकाइयां काम करती हैं 8086 को एक अतिव्यापी निर्देश लाने और निष्पादन तंत्र देने के लिए अतुल्यकालिक रूप से जिसे पिपेलिनिंग कहा जाता है। इसके परिणामस्वरूप सिस्टम बस और सिस्टम का कुशल उपयोग होता है प्रदर्शन। बीआईयू में निर्देश कतार, खंड रजिस्टर, निर्देश सूचक शामिल हैं, पता जोड़ने वालायूरोपीय संघ में नियंत्रण सर्किटरी ., निर्देश डिकोडर, एएलयू, पॉइंटर और इंडेक्स शामिल हैं रजिस्टर, ध्वज रजिस्टर।

बस इंटरफैकर यूनिट:

यह १६ बिट कि दो दिशाओ वाली तथा २० बिट कि एड्रेस बस प्रोविड करता है यह| | सभी बाहरीबस प्रचालन के लिए जिमेदार होता है

ये निम्न फलन विशेष रूप से रखता है :

यह कतार निर्देश कोड के सिक्स बाइट्स तक की प्रीफेच की अनुमति देती है। जब भी कतार लगी हो बीआईयू भरा नहीं है, इसमें कम से कम दो और बाइट्स के लिए जगह है और एक ही समय में यूरोपीय संघ है मेमोरी से ऑपरेंड पढ़ने या लिखने का अनुरोध नहीं करते हुए, बीआईयू आगे देखने के लिए स्वतंत्र है अगले अनुक्रमिक निर्देश को पूर्व निर्धारित करके कार्यक्रम। ये प्रीफ़ेचिंग निर्देश वे अपने FIFO कतार में आयोजित किए जाते हैं। अपने 16 बिट डेटा बस के साथ, बीआईयू दो निर्देश बाइट्स प्राप्त करता है एक एकल स्मृति चक्र। कतार के इनपुट छोर पर एक बाइट लोड होने के बाद, यह स्वचालित रूप से FIFO के माध्यम से आउटपुट के निकटतम खाली स्थान पर स्थानांतरित हो जाता है।

कार्यान्वयन इकाई आउटपुट अंत से कतार तक पहुंचता है। यह एक के बाद एक निर्देश बाइट पढ़ता है कतार के आउटपुट से। यदि कतार भरी हुई है और यूरोपीय संघ पहुंच का अनुरोध नहीं कर रहा है स्मृति में संचालन। कोई बस गतिविधि नहीं के ये अंतराल, जो बस चक्रों के बीच हो सकते हैं निष्क्रिय अवस्था के रूप में जाना जाता है। यदि बीआईयू

पहले से ही एक निर्देश प्राप्त करने की प्रक्रिया में है जब कार्यान्वयन इकाई रीड तथा राईट ओप्रन्ड्स को मेमोरी तथा इनपुट| आउटपुट से अनुरोध करता है/

8086 के आंतरिक रजिस्टर

8086 में उपयोगकर्ता के चार समूह सुलभ आंतरिक रजिस्टर हैं। वे निर्देश हैं पॉइंटर, चार डेटा रजिस्टर, चार पॉइंटर और इंडेक्स रजिस्टर, चार सेगमेंट रजिस्टर। 8086 में कुल चौदह 16-बिट रजिस्टर हैं, जिसमें 16 बिट रजिस्टर शामिल है जिसे स्टेटस कहा जाता है स्थिति और नियंत्रण झंडे के लिए लागू बिट्स में से 9 के साथ रजिस्टर करें। अधिकांश रजिस्टर 64 KB मेमोरीसेगमेंट के भीतर डेटा-चार अलग .अनुदेश ऑफ़सेट शामिल हैं/ अलग64 हैं निर्देश, स्टैक, डेटा और अतिरिक्त डेटा के लिए KB सेगमेंट। यह निर्दिष्ट करने के लिए कि I MB में कहां है प्रोसेसर मेमोरी ये 4 सेगमेंट स्थित हैं प्रोसेसर चार सेगमेंट रजिस्टरों का उपयोग करता है:

कोड सेगमेंट (CS)

एक 16-बिट रजिस्टर है जिसमें प्रोसेसर के साथ 64 KB सेगमेंट का पता होता है |निर्देश प्रोसेसर द्वारा संदर्भित निर्देशों के लिए सभी एक्सेस के लिए सीएस सेगमेंट का उपयोग करता है अनुदेश सूचक रजिस्टर। सीएस रजिस्टर सीधे नहीं (आईपी)बदला जा सकता है।

स्टैक सेगमेंट:

यह एक १६ बिट का रजिस्टर है जो एड्रेस को ६४ के बी में प्रोग्राम स्टैक के साथ रखता है |

सीएस रजिस्टर

8086 माइक्रोप्रोसेसर के सभी सामान्य रजिस्टरों का उपयोग अंकगणित और तर्क के लिए किया जा सकता है कार्रवाई। सामान्य रजिस्टर हैं:

संचायक रजिस्टर में दो 8-बिट रजिस्टर एएल और एएच होते हैं, जिन्हें जोड़ा जा सकता है एक साथ और 16-बिट रजिस्टर एएक्स के रूप में उपयोग किया जाता है। इस मामले में एएल में निम्नक्रम बाइट शामिल है-

शब्द, और एएच में उच्च क्रम बाइट शामिल है। संचायक का उपयोग आई ओ संचालन / के लिए किया जा सकता है और स्ट्रिंग हेरफेर।

आधार रजिस्टर में दो 8-बिट रजिस्टर बीएल और बीएच होते हैं, जिन्हें एक साथ जोड़ा जा सकता है और 16-बिट रजिस्टर बीएक्स के रूप में उपयोग किया जाता है इस . क्रम बाइट शामिल है-मामले में बीएल में शब्द का निम्न, और बीएच में उच्च क्रम बाइट- शामिल है। बीएक्स रजिस्टर में आमतौर पर एक डेटा पॉइंटर होता है जिसका उपयोग किसके लिए किया जाता है? आधारित, आधारित अनुक्रमित या रजिस्टर अप्रत्यक्ष पता। काउंट रजिस्टर में दो 8-बिट रजिस्टर सीएल और सीएच होते हैं, जिन्हें एक साथ जोड़ा जा सकता है और 16-बिट रजिस्टर सीएक्स के रूप में उपयोग किया जाता है। संयुक्त होने पर, सीएल रजिस्टर में कमक्रम बाइट होता है-

अतिप्रवाह ध्वज)- सेट करें यदि परिणाम बहुत बड़ी सकारात्मक संख्या है, या बहुत छोटा नकारात्मक है गंतव्य ऑपरेंड में फिट होने के लिए नंबर।

दिशा ध्वज)DF) - यदि सेट किया जाता है तो स्ट्रिंग हेरफेर निर्देश स्वत घटाने : सूचकांक होगा रजिस्टर। यदि इसे मंजूरी दे दी जाती है तो सूचकांक रजिस्टर स्वत वेतन : वृद्धि होगी।

इंटरप्ट इस बिट को सेट करना मास्केबल इंटरप्ट को सक्षम - (आईएफ) इनेबल फ्लैग बनाता है।

सिंगलयदि सेट किया जाता है तो अगले नि - (टीएफ) स्टेप फ्लैग-र्देश के बाद सिंगल- टेप इंटरप्ट होगा।ंस साइन फ्लैग यदि परिणाम का सबसे महत्वपूर्ण बिट सेट - (एसएफ) किया गया है तो सेट करें।

शून्य ध्वज)ZF) - यदि परिणाम शून्य है तो सेट करें

सहायक कैरी फ्लैग सेट करें यदि एएल में बिट्स - (एएफ)0-3 से कोई कैरी या उधार लिया गया था पंजीकृत करें।

समता ध्वज)PF) - यदि परिणाम के निम्न") क्रम बाइट में समता-l" बिट्स की संख्याक्या है (?

यहाँ तक कि। कैरी फ्लैग सेट करें कि क्या सबसे महत्वपूर्ण बिट से कोई कैरी - (सीएफ) या उधार लिया गया था अंतिम परिणाम गणना।

एड्रेसिंग मोड

विस्थापन के साथ अनुक्रमित आधारित - :8-बिट या 16-बिट निर्देश संचालन में जोड़ा जाता है आधार रजिस्टर (एसआई या डीआई) और इंडेक्स रजिस्टर (बीएक्स या बीपी) की सामग्री, परिणामी मान क्या है? उस स्थान को इंगित करें जहां डेटा रहता है।

बाधाएं

प्रोसेसर में निम्नलिखित बाधाएं हैं:

INTR एक मास्केबल हार्डवेयर इंटरप्ट है। इंटरप्ट का उपयोग करके सक्षम अक्षम किया जा सकता है एसटीएल सीएलआई निर्देश या झंडे रजिस्टर को अपडेट करने की अधिक / जटिल विधि का उपयोग करना पीओपीएफ निर्देश की मदद से।

जब कोई व्यवधान होता है, तो प्रोसेसर फ्लैग्स रजिस्टर को स्टैक में संग्रहीत करता है, आगे अक्षम करता है व्यवधान, बस से एक बाइट आता है जो इंटरप्ट प्रकार का प्रतिनिधित्व करता है, और बाधित करने के लिए कूदता है

प्रसंस्करण नियमित पता जिसका स्थान interrupt typo में संग्रहीत किया जाता है। काटना

प्रसंस्करण दिनचर्या आईआरईटी निर्देश के साथ वापस आनी चाहिए।

एनएमआई एक नॉन मास्केबल इंटरप्ट है। इंटरप्ट को उसी तरह संसाधित किया जाता है-

काटना। एनएमआई का इंटरप्ट प्रकार 2 है, यानी एनएमआई प्रोसेसिंग रूटीन का पता क्या है?

पता

003FFH

उपलब्ध हस्तक्षेप

003FCH

वैक्टर)224)

आरक्षित हस्तक्षेप

वैक्टर)27)

00014H

00010H

समर्पित

काटना

OOOOCH

वैक्टर)5)

OOOOBH

00008H

00007H

00004H

00003H

00002H

00001 H

OOOOOH

टाइप एफएफएच वेक्टर (उपलब्ध)

टाइप 21 एच वेक्टर (उपलब्ध)

टाइप 20एच वेक्टर (उपलब्ध)

टाइप करें IFH वेक्टर (आरक्षित)

टाइप 05एच वेक्टर (आरक्षित)

टाइप 04एच वेक्टर (अतिप्रवाह)

टाइप 03 एच वेक्टर

(आई(बाइट आईएनटी निर्देश-

टाइप 02एच वेक्टर (एनएमआई)

टाइप OIH वेक्टर

(जाल या एकल कदम(

टाइप OOH वेक्टर

(विभाजन(ओ त्रुटि-दर-

8 बिट्स

8086 में इंटरप्ट वेक्टर तालिका

न्यूनतम मोड इंटरफ़ेस

जब न्यूनतम मोड कार्रवाई का चयन किया जाता है, तो 8086 सभी नियंत्रण संकेत प्रदान करता है

मेमोरी और वीओ इंटरफ़ेस को लागू करने की आवश्यकता है। न्यूनतम मोड सिग्नल क्या हो सकता है?

निम्नलिखित बुनियादी समूहों में विभाजितडेटा बस / पता :, स्थिति, नियंत्रण, हस्तक्षेप और

डीएमए।

डेटा बस ये लाइनें दो कार्य करती हैं। एक पते के रूप में बस :20 बिट लंबी है और

पहले मशीन चक्र के दौरान और अगले मशीन चक्रों के दौरान डेटा बस के रूप में। D15 MSB है

और डीओ एलएसबी। डेटा बस के रूप में कार्य करते समय, वे मेमोरी के लिए डेटा पढ़ते लिखते हैं /,

I/O उपकरणों के लिए इनपुटआउटपुट डेटा/, और एक इंटरप्ट नियंत्रक से प्रकार कोड को बाधित करें।

काटना

रीसेट

HI-DA

एमएन, 'निक्स'

Do-DE

बीएचई, 'एस,

तैयार

LO नियंत्रण

न्यूनतम मोड 8086 WU का ब्लॉक आरेख

स्थिति संकेत चार सबसे महत्वपूर्ण पता लाइनें :A19 से A16 तक भी हैं

मल्टीप्लेक्स लेकिन इस मामले में स्थिति के साथ एस 6 से एस 3 के माध्यम से संकेत मिलता है। ये स्थिति बिट्स आउटपुट हैं

बस पर एक ही समय में डेटा अन्य बस लाइनों पर स्थानांतरित किया जाता है। बिट S4 और S3

एक साथ 2 बिट बाइनरी कोड से जो पहचानता है कि 8086 आंतरिक खंड में से कौन सा पंजीकृत है

इसका उपयोग उस भौतिक पते को उत्पन्न करने के लिए किया जाता है जो यात्रा के दौरान पता बस पर आउटपुट किया गया था

वर्तमान बस साइकिल। कोड S4S3 00 एक रजिस्टर की पहचान करता है जिसे अतिरिक्त सेगमेंट रजिस्टर के रूप में जाना जाता है

सेगमेंट पते का स्रोत.

सेगमेंट रजिस्टर

अतिरिक्त

स्टैक

कोड कोई नहीं /

D ata

s,

1

1

1

o

नियंत्रण संकेत नियंत्रण संकेत :8086 मेमोरी I / O का समर्थन करने के लिए प्रदान किए जाते हैं इंटरफ़ेस। वे कार्यों को नियंत्रित करते हैं जैसे कि बस को एक वैध पता ले जाना है जिसमें दिशा डेटा को बस पर स्थानांतरित किया जाना है, जब वैध लेखन डेटा बस पर होते हैं और सिस्टम बस पर रीड डेटा कब डालना है।

ALE तर्क I की एक नाड़ी है जो बाहरी सर्किटरी को संकेत देता है जब एक वैध पता शब्द पर होता है बस। इस पते को पल्स के आई ओ किनारे पर बाहरी सर्किटरी में-टू-बैंक उच्च सक्षम है।

इस पर लॉजिक ओ का उपयोग मेमोरी के रूप में किया जाता है जो डेटा के सबसे महत्वपूर्ण बाइट आधे हिस्से के लिए सिग्नल सक्षम करता है

बस D8 के माध्यम से DI. ये लाइनें एक दूसरा फ़ंक्शन भी प्रदान करती हैं, जो एस 7 स्टेटस लाइन के रूप में है। M/IO और DT/R लाइनों का उपयोग करके, 8086 संकेत देता है कि किस प्रकार का बस चक्र प्रगति पर है और किस दिशा में डेटा को बस पर स्थानांतरित किया जाना है।

अधिकतम मोड इंटरफ़ेस

जब 8086 अधिकतममोड कॉन्फ़िगरेशन के लिए सेट किया जाता है-, तो यह इसके लिए सिग्नल प्रदान करता है एक मल्टीप्रोसेसर कोप्रोसेसर सिस्टम वातावरण को लागू / करना। द्वारा बहुप्रोसेसर पर्यावरण हमारा मतलब है कि सिस्टम में एक माइक्रोप्रोसेसर मौजूद है और प्रत्येक प्रोसेसर अपने स्वयं के कार्यक्रम को निष्पादित कर रहा है। आमतौर पर इस प्रकार के सिस्टम वातावरण में, कुछ होते हैं सिस्टम संसाधन जो सभी प्रोसेसर के लिए सामान्य हैं। उन्हें वैश्विक संसाधन कहा जाता है।

ऐसे अन्य संसाधन भी हैं जो विशिष्ट प्रोसेसर को सौंपे गए हैं। इन्हें किस रूप में जाना जाता है? स्थानीय या निजी संसाधन। कोप्रोसेसर का मतलब यह भी है कि इसमें एक दूसरा प्रोसेसर है प्रणाली। इसमें दो प्रोसेसर एक ही समय में बस तक नहीं पहुंचता है। एक पास हो जाता है सिस्टम बस का नियंत्रण दूसरे पर और फिर इसके संचालन को निलंबित कर सकता है। अधिकतम में-

मोड 8086 प्रणाली, वैश्विक संसाधनों के आवंटन को लागू करने के लिए सुविधाएं प्रदान की जाती हैं और अन्य माइक्रोप्रोसेसर या कोप्रोसेसर को बस नियंत्रण पास करना।

8288 द्वारा उत्पादित नियंत्रण आउटपुट DEN, DT / R और ALE हैं। ये 3 संकेत न्यूनतम सिस्टम मोड के लिए वर्णित कार्यों के समान कार्य प्रदान करें। का यह सेट

बस कमांड और नियंत्रण सिग्नल मल्टीबस और उद्योग मानक के साथ संगत है

माइक्रोप्रोसेसर सिस्टम को इंटरफेस करने के लिए।

8289 बस मध्यस्थ बस -मध्यस्थता और लॉक सिग्नल:

यह डिवाइस प्रोसेसर को सिस्टम बस पर रहने की अनुमति देता है। यह इसे लागू करके ऐसा करता है 8086-आधारित प्रणाली में मल्टीबस मध्यस्थता प्रोटोकॉल। 8288 बस नियंत्रक के अतिरिक्त

और 8289 बस मध्यस्थ नियंत्रण संकेतों का उत्पादन करने के लिए उपयोग के लिए 8086 पिनों की एक संख्या को मुक्त करता है कई प्रोसेसर का समर्थन करने के लिए आवश्यक हैं। बस प्राथमिकता लॉक इन संकेतों में से एक है। (लॉक) यह एस 2 के माध्यम से स्थिति संकेतों एसओ के साथ बस मध्यस्थ को इनपुट है।

कतार स्थिति सिग्नलदो नए सिग्नल : जो अधिकतम में 8086 द्वारा उत्पादित होते हैं-

मोड सिस्टम क्यूएसओ और क्यूएसआई क्यूएसआई क्यू स्थिति आउटपुट हैं। साथ में वे 2-बिट कतार बनाते हैं

8. 68000 माइक्रोप्रोसेसर

8.1 परिचय (Introduction)

68000 मोटरोला श्रेणी का एक 16 बिट का माइक्रोप्रोसेसर है इसके पश्चातय इसकी कई श्रेणी आयी जिनमें 68008, 68010 68020 तथा 68030 आदि प्रमुख है। 68000 में 24 बिट की एड्रेस बस होती है तथा 16 बिट की डाटा बस होती है अथार्त 32 बिट का डाटा आ भी सकता है और इससे जा भी सकता है। यही कारण है कि यह 32 बिट का माइक्रोप्रोसेसर भी कहलाता है।

68000 को SV सप्लाई की आवश्यकता होती है तथा इसकी आन्तरिक -क्लॉक आवृती 25 MHz होती है। इस माइक्रोप्रोसेसर में सीधे एड्रेस की जाने वाली मेमोरी 16 MB की होती है। 68K K किलो-1000) माइक्रोप्रोसेसर में चिप पर कोई क्लॉक परिपथ नही होती है। इसलिये इसमें एक क्रिस्टल प्रवर्धक या वाह्य क्लॉक जेनेरेटर परिपथ लगे हुये होते हैं।

68000, चालन यंत्र बचाव को लगाने के लिये,

यह दो मोड पर कार्य करता है।

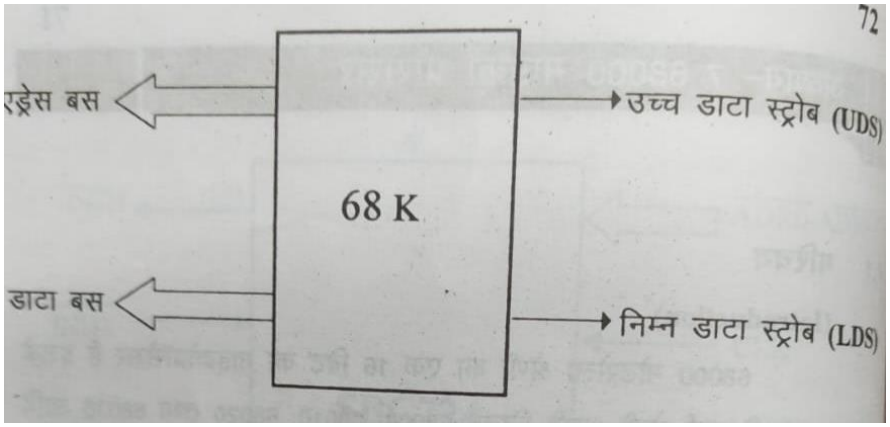
निरीक्षक मोड (Supervisor Mode)

उपयोग कर्ता मोड (User Mode)

निरीक्षक मोड, प्रबंध तंत्र मोड भी कहलाता है। इन मोड में 68000 माइक्रोप्रसेसर अपने समस्त निर्देशों को हल करता है।

7.2 मेमोरी संगठन (Memory Organisation)

जैसा चित्र (7.1) में विदित है कि 68000 माइक्रोप्रोसेसर ऐड्रेस बस. टा बस तथा स्ट्रोब लाइन का इस्तेमाल करके एक मेमोरी बाइट को बताता है।



चित्र (7.1) ब्लाक मानचित्र 68 K

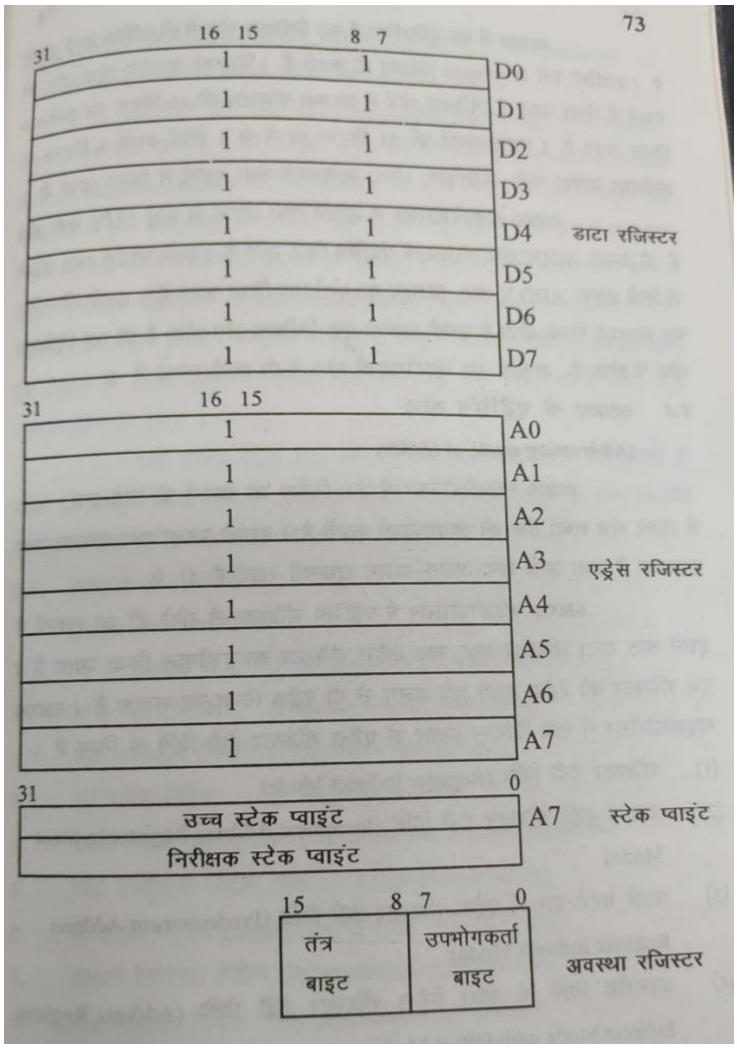
यहाँ स्ट्रोब डाटा को दो तरह से उपयोग में लाया जाता है। उच्च डाटर (Upper Data Strobe) तथा निम्न डाटा स्ट्रोब (Lower data strobe) जब एक पूरा शब्द ऐड्रेस किया जाता है तो यह दोनो स्ट्रोब निम्न क्रियाशील होता है जो बात पर निर्भर करता है कि उच्च बाइट या निम्न बाइट क्रियाशील हो रही है। 6000 में 24 जिन क्रियाशील होती है जिसमें 23 पिन ऐड्रेस के लिए होती है तथा शेष एक पिन उच्च डाटा स्ट्रोब (UDS) तथा निम्न डाटा स्ट्रोब (LDC) के अवस्था संकेतों के लिए होती है।

8.3 रजिस्टर संगठन (Register Organisation)

68000 माइक्रोप्रोसेसर में 32 बिट के रजिस्टर होते हैं जिनका इस्तेमाल कई बीजगणितीय तथा लॉजिक प्रचालनों के लिये किया जाता है। यह रजिस्टर एक शब्द या बड़े शब्दों की तरह भी उपयोग में लाये जा सकते हैं। 68000 माइक्रोप्रोसेसर में 32

बिट के आठ डाटा रजिस्टर तथा 32 बिट के सात एड्रेस रजिस्टर होते हैं। यह सात एड्रेस रजिस्टर आपरेंड के एड्रेस को बताने के लिये होते हैं चित्र (8.2)

इसके अलावा 32 बिट के दो स्टेक प्वाइंटर होते हैं जो इस बात पर निर्भर करते हैं कि माइक्रोसेसर किस मोड में प्रचलित हैं। जैसा कि पहले हम देख चुके हैं कि निरीक्षक मोड तथा उपयोगकर्ता मोड पर कार्य करता है। एक समय में सिर्फ एक ही मोड क्रियाशील होता है। 68000 माइक्रोप्रोसेसर में 32 बिट का एक



68000 में यह सुनिचित है यह निरीक्षक मोड में ही अधिक कार्य है। इसलिये इसे Privileged उवक्रम भी कहते हैं। जिसका उपयोग स्टे देखने में किया जाता है निरीक्षक मोड में अवस्था रजिस्टर की 16 बिट्स का इस किया जाता है। उपयोगकर्ता की 81

बिट्स का में से 1 सिर्फ प्रथम 4 दिस इस्तेमाल क्रमश: केरी, ओवरफ्लो, जीरो, ऋणात्मक तथा बढ़ाने में किया जाता है।

68000 माइक्रोप्रोसेसर में जोड़ने तथा घटाने के कोई निर्देशन हैं जो क्रमश: ADDC तथा SUBC में प्रदर्शित किये जाते हैं। इनमें जोड़ने के लिये क्रमशः ADDX तथा SUBX का इस्तेमाल किया जाता है । इससे तीन है का रूकावट मास्क होता है इसके अलावा एक निरीक्षक लेग होता है तो यह ि मोड में होता है अन्यथा यह उपयोगकर्ता मोड में ही कार्य करता है ।

8.4 68000 के एड्रेसिंग मोड (Addressing mode of 68000)

68000 माइक्रोप्रोसेसर में एक निर्देश को चलाने के लिये एक से लेकर पांच शब्दों तक की आवश्यकता पड़ती है। इसका पहला शब्द प्रचलन ह कहलाता है तथा अन्य शब्द अलग-अलग सूचनायें रखते हैं ।

68000 माइक्रोप्रोसेसर में एड्रेसिंग रजिस्टर से सीधे की जा सकती है इसमें आठ डाटा रजिस्टर तथा सात ऐड्स रजिस्टर का इस्तेमाल किया जाता है। इसे रजिस्टर को ऐड्रेस करते हुये अलग से भी एड्रैस किया जा सकता है 1 6600 माइक्रोप्रोसेसर में पांच विभिन्न प्रकार के एड्स रजिस्टर टेढ़ी विधि के निम्न है

(1) रजिस्टर टेढी विधि (Register Indirect Mode)

(2) बाद में वृद्धी रजिस्टर टेढी विधि (Increment Address Register Indirect Mode)

(3) पहले घटते क्रम में एड्रेस रजिस्टर टेढ़ी विधि (Predecrement Address Register Indirect Mode)

(4) आफसेट विधि के साथ ऐड्रेस रजिस्टर टेढ़ी विधि (Address Reg Indirect Mode with Offset Mode)

(5) चिन्ह के साथ ऐड्रेस रजिस्टर टेढी विधि, ऑफसेट के साथ (Indexed Register Indirect Mode with Offset Mode)

रजिस्टर टेढी विधि (Register indirect addressing) एड्रेस रजिस्टर प्रभावी एड्स को रखता है। बाद में वृद्धी ऐड्रेस रजिस्टर टेडी विधि (Post evicroment indirect addressing mode) में ऐड्रेस रजिस्टर को 1, 2 तथा 4 के क्रम में बढ़ाया जा सकता है। यदि एक बाइट है तो 1 के क्रम में, एक शब्द है तो 2 के क्रम में तथा एक लम्बा शब्द है तो 4 के क्रम में बढाया जा सकता है। इसी तरह पहले घटते क्रम में ऐड्स रजिस्टर टेढी विधि (Prederement address register indirect mode) में एक रजिस्टर को उसके उपयोग करने से पहले 1. 2 तथा 4 के क्रम में घटाया (decrement) जा सकता है। यह घटाव एक बाइट एक शब्द तथा एक लम्बे शब्द के लिये 1. 2 तथा 4 क्रम का होगा।

इसी प्रकार ऐड्रेस टेडी विधि में ऑफसेट तथा चिन्ह तालिकाओं के प्रवेश को स्वकृत किया गया है। ऑफसेट विधि में प्रभावी रजिस्टर के साथ 16 बिट का एक चिन्ह पूर्णांक को जोड़ा दिया जाता है।

8.5. 68000 के निर्देश (Instruction of 68000)

68000 के निर्देशों को आठ भागों में बाटा जाता है।

1. डाटा चाल निर्देश (Data Movement Instruction)

2. अंक गणितीय निर्देश (Arithmatic Instruction)

3.लॉजिकल निर्देश (Logical Instruction)

4. स्थान बदल तथा घुमाव निर्देश (Shift and Rotate Instruction)

5. बिट दस्तकारी निर्देश (Bit manipulation Instruction)

6. बी.सी.डी. निर्देश (Binary Coded decimal Instruction)

7. प्रोग्राम नियंत्रित निर्देश (Program control Instruction)

8. तंत्र नियंत्रित निर्देश (System control Instruction)

डाटा चाल निर्देश में डाटा को कई तरह से अन्तरण किया जा सकता है डाटा रजिस्टर से रजिस्टर में रजिस्टर से मेमोरी में मेमोरी से रजिस्टर में त से मेमोरी में अन्तरण किया जा सकता है।

68000 माइक्रोप्रोसेसर में कई तरह के चाल निर्देश होते हैं यदि दो रजिस्टर के डाटा को आपस में बदलना हो तो EXG Rx, Ry निर्देश इस्तेमाल किया जाता है। जहां EXG Exchange को बताता है।

यदि किसी प्रभावी रजिस्टर को एक निश्चित ऐसिंग विधि में ग किया गया है तथा फिर इसे एक ऐड्स रजिस्टर में रखा जाता है तो इसे निम्न निर्देशित किया जायेगा।

LEA (EA), An

यदि हम 68000 माइक्रोप्रोसेसर पर इन चाल निर्देशों को देखे तो Move Command के अन्तर्गत आदेशित होता है। चाल आदेशों को दो तरह से देख जाता है। पहला ब्लॉक चाल (Move Block) इसमें एक मेमोरी जगह, दूसरी मेमोरे जगह में चली जाती है। इसी तरह दूसरा आदेश होता है स्थिरांक चाल (Mo Constant)

इसमें एक दी हुई मेमोरी जगह से स्थिर डाटा को लोड करता है। य डाटा 2 अंकों का हेक्साडेसीमल डाटा होता है ।

अंक गणितीय निर्देश में 68000 माइक्रोप्रोसेसर 8 16 तथा 32 बिट डाटा का जोड़ तथा घटाना करता है। दो 16 बिट्स को गुणनफल निकाल है । 32 बिट के डाटा को 16 बिट के द्वारा भागित किया जाता है। इसके अलाव निर्देशों को आपस में मिलाना उन्हें साफ करना तथा अस्वीकार करने का कार्य भी करता है ।

जोड़ने के लिये निम्न निर्देश इस्तेमाल में लाये जाते हैं ।

ADD

ADDI - तुरंत जोड़ने के लिये

ADDQ - दिये गये ऐड्स के डाटा के साथ 0 से 7 तक के अंको को जोड़ना ।

इसी तरह घटाने के लिये

SUB

SUBI

SUBQ

गुणा करने के लिये

MULS

MULS- चिन्ह रहित गुणनफल के लिये

भाग करने के लिये

DIVS

DIVS- चिन्ह रहित भाग के लिये

निर्देशों के मिलान के लिये

CMPM

सभी बिट्स को साफ (Clear) करने के लिये अर्थात करने के लिये

CLR.L

अस्वीकार करने के लिये

NEG.W

जहां तक निर्देशों को मिलाने (Compare) है यह 68000 माइक्रोप्रोसेस में तीन तरह के आदेशों के द्वारा किया जाता है।

ब्लॉक मिलान (Clock Compare)

स्थिरॉक मिलान (Constant Compare)

बाइट मिलान (Byte Compare)

लॉजिकल निर्देशों (Logical Instructions) में AND, OR और NOT के अलावा OR को भी इस्तेमाल किया जाता है। इसमें निम्न निर्देशों को इस्तेमाल किया है |

AND (EA), (EA)

ANDI # data, (EA)

प्रोग्राम गणक होता है जो इस बात को निर्धारित करता है कि माइक्रोप्रोसेसर में किस क्रम में निर्देश हल होते हैं ।

68000 माइक्रोप्रोसेसर में अवस्था रजिस्टर भी होते हैं जो दो भागों में बटा होता है। पहली उपभोग कर्ता बाइट तथा दूसरा निरीक्षक बाइट यहाँ महत्वपूर्ण बात यह है कि उपभोगकर्ता मोड में निरीक्षक बाइट को परिवर्तित नहीं किया जा सकता है।

ANDI # data, (CCR)

ANDI # data 16, SR

EOR Dn, (EA)

68000 माइक्रोप्रोसेसर में बहुत से ऐस निर्देश आदेश होते हैं जो अंकगणितीय प्रचलन में काफी सहायक होते हैं। महत्वपूर्ण बात यह होती है कि इस घुमाव में चिन्ह बिट का स्थान नही बदलता है।

उदाहरण के लिये यदि ASL, Dx, DY निर्देश का इस्तेमाल किया जाता है तो यह Dy का बायी तरफ सरकायेगा जो Dx में विशिष्ट वर्णन करने योग्य है। यह Dx की निम्न बिट का 0-63 बार स्थान बदलेगा। यदि निर्देश ASL# data dx हैं तो विधि वही होगी जो अभी उपर वर्णित की है परन्तु यह 0-7 वार ही तुरंत डाटा का स्थान बायी और बदलेगा। इसी तरह दायी ओर स्थान बदलने के लिये निम्न निर्देश का इस्तेमाल किया जाता है।

ASR Dx, Dy ASr# data Dx

ASR (EA)

Rol Dx, Dy- इसमें Dx की 6 निम्न बिट की संख्या Dx को स्थान बदलती है।

बिट दस्तकारी निर्देश में एक विशिष्ट वर्णित बिट का 1.5 काम्पलीमेट, 2 पलेग में दिखाई देता है। यह 0 को साफ करता है तथा 1 में बदलता है या उन्हें बिना - बदले ही रहने देता है। इसमें जिन निदशों की आवश्यकता होती है वे निम्न है

BCHG

BCLR

BSET

BTST

बाइनरी कोड डेसीमल निर्देश में निर्देश सेट तीन BCD निर्देशों को रखता है।

ABCD - जोड़ने के लिए (For adding)

SBCD - घटाने के लिए (For Subtraction)

NBCD - अस्वीकार करने के लिए (For Negation)

प्रोग्राम नियंत्रित निर्देश जिसमें ब्रांच (Branch), कूद (Jump) तथा सबरूटीन (Subroutine) कॉल निर्देशित हैं उनको संपादित करने के लिए 14 निर्देश है-

BCC- BGT

BCS-BGE

BEQ-BLE

BNE-BLT

BHI-BMI

BLS-BVC

BVC

तंत्र नियंत्रक निर्देश जो 68000 माइक्रोप्रोसेसर के नियंत्रक होते हैं। इन के साथ दो अन्य निर्देश होते हैं ट्रेप (Trap) तथा चेक (Check) निर्देश होते हैं। तंत्र नियंत्रक निर्देशों में कुछ महत्वपूर्ण निर्देश निम्न हैं ।

RESET निर्देश

निरीक्षण मोड में जब यह निर्देश दिया जाता है तो एक बाह्य संकेत उत्पन्न होता है जो 68000 से जुड़ी युक्तियों को प्रारंभिक अवस्था में कर देता है । 68000 माइक्रोप्रोसेसर में RESET पिन द्विदिशा की होती है। जिसे हार्डवेयर और साफ्टवेय द्वारा उपयोग किया जा सकता है ।

Move USP, (An) or Move (An) USP

USP (उच्च स्टेक प्वाइंटर) का उपयोग रखने (Save) बनाये रखने (Restore)

तथा निरीक्षण मोड में उच्च स्टेक प्वांइटर की सूची को बदलने में किया जाता है।

TRAP निर्देश

68000 माइक्रोप्रोसेसर में 16 Trap निर्देश होते हैं जिन्हें 0 से 15 तक रखा जाता है । इन्हे सदिश हेक्साडेसीमल एड्रेस द्वारा इंगित किया जाता है ।

9. 486 पेंटियम तथा पेंटियम प्रो माइक्रोप्रोसेसर में अंकगणितीय तथा लॉजिकल निर्देश

9.1 परिचय (Introduction):

486 पेटीयम तथा पेटीयम प्रो माइक्रोप्रोसेसर अंकगणितीय निर्देश प्रचलनों को हल करने के लिये होते हैं।

जोड़ना (Addition)

घटाना (Subtraction)

गुणनफल (Multiplication)

तुलना करना (comparison)

अंकों के बढ़ते तथा घटते क्रम (Increment and decrement)

इसी तरह लॉजिकल निर्देशों में निम्न प्रचलनों को देखना होता है।

AND

OR

EX-OR

NOT

SHIFT

PROTATE

लॉजिकल तुलना (Logical Comparison)

9.2. जोड़ना (Addition)

486, पेटीयम तथा पेटीयम प्रो माइक्रो प्रासेसर में ADD निर्देश उपयोग किया जाता है। यह निर्देश 8 बिट, 16 बिट तथा 32 बिट की बाइन संख्याओं को जोड़ता है ।

इन माइक्रोप्रोसेसर में बाइनरी संख्याओं को कई तरह से जोड़ा जाता है।

9.2.1 मेमोरी से रजिस्टर का जोड़ना (Memory to Register addition)

इस तरह के जोड़ में मेमोरी के डाटा तथा रजिस्टर के डाटा जसे पहुंच चिन्ह रजिस्टर (Destination Index register) के साथ जोड़ा जाता है । इसे निम्न उदाहरण द्वारा समझा जा सकता है।

```
Mo DI OFFSET NUMB
MOVAL O
ADD AL, (DI)
ADD AL, (DI+ 1)
RET
```

9.2.2 रजिस्टर जोड (Register Addition)

इस निर्देश में दो रजिस्टरों को जोड़ा जाता है

उदाहरण के लिये

```
ADD AX, BX
```

ADD AX, CX

ADD AX, DX

9.2.3 तुरन्त जोड़ (Immediate Addition)

इस तरह के जोड़ में स्थिरॉक तथा डाटा जो हमे मालूम हैं उन्हें जोड़ा जाता है। उदाहरण के लिये

MOV DL, 12H

ADD DL, 33H

यह पहले move निर्देश के द्वारा DL को 12H के साथ लोड किया जाता है तथा फिर इसमें 33H जोड़ दिया जाता है।

9.2.4 क्रमानुसार जोड़ना (Array Addition)

जैसा कि हम जानते हैं कि मेमोरी का एक क्रम डाटा की श्रेणी बताता है। डाटा का एक क्रम हुये होते हैं

ADDAL, ARRAY (SI))- Add element 3

ADDAL, ARRAY (SI +2) – Add element 5

ADDAL ARRAY (SI +4) - Add element 7

यहाँ SI जो एक एड्रेस तत्व को निर्देशित करता है तथा एक ि क्रम में इसे AL के साथ जोड़ता जाता है। 80386 में डाटा का एक क्रम 16 बिट रखता है और उपयोग के बाद 16 बिट को जोड़ बनाता है तथा इसे AX रजिस्टर रखता है।

इसके अलावा इन माइक्रोप्रोसेसर में दो निर्देश होते हैं। EBX एड्रेस क्रम को लोड करता है तथा ECX जो क्रम तत्व नम्बर को रोक रखता है।

9.2.5 वृद्धि जोड़ (Increment Addition)

इसमें INC निर्देश का इस्तेमाल किया जाता है जो एक को किसी एक रजिस्टर या मेमोरी जगह में जोड़ देता है। यदि PIR WORD. PTR निर्देश इस्तेमाल किया गया है तो डाटा के आकार को बताना अति आवश्यक होता है। D रजिस्टर की सूची को एक से बढ़ाना

INC DI , D रजिस्टर की सूची को एक से बढ़ाना

INC BL, BL=BL + 1,

INC BYTE PTR (BX)- मेमोरी जगह जो BX के द्वारा एड्रेस की गई है उस बाहर में एक जोड़न

INC WORD PTR (SI)- मेमोरी जगह की सूची जो SI एड्रेस की गई है उसे डाटा में एक जोड़ना

INC DWORD PTR (ECX)- मेमोरी जगह जहां दुगना शब्द है तथा जो ECX द्वारा निर्देशित है उसमें एक जोड़ना

INC DATA1- मेमोरी जगह DATAI को एक से जोड़ना

9.2.6 केरी के साथ जोड़ना (Addition with carry)

इन माइक्रोप्रोसेसर में किसी भी जोड़ जिससे हॉसिल (Carry) आती हो उसके साथ भी जोड़ा जा सकता है। इसके लिये ADC निर्देश का उपयोग किया जाता है ।

ADDC AL, AH - AL+AH+CARRY

ADCCX, BX- CX+BX+CARRY

ADC EBX, EDX - EBX= EBX+EDX+CARRY

ADC AH, (CX)- मेमोरी जगह जो एक बाइट रखती है तथा CX के द्वारा एड्रेस की गई है उसे AH के साथ जोड़ना केरी भी जोड़ना तथा परिणाम को AH में रखना।

ADC CX, (BP+2)- एक शब्द जो स्टेक भाग की पंक्ति है मेमोरी जगह BP+2 द्वारा एड्रेस की गई है को केरी के साथ CX में जोड़ना तथा परिणाम को CX में रखना ।

9.3 घटना (Subtraction)

इसमें SUB निर्देश का उपयोग किया जाता है। जिस तरह जोड़ा कई तरह से जाता है। दो रजिस्टर को आपस में घटाया जा सकता है जिससे रजिस्टर घटाना (Register Subtraction) कहते हैं। तुरंत घटाना भी किया जा सकता है । अंक गणितीय घटाने में जिस तरह एक अंक को दस मान के घटाया जाता है उसी तरह इन माइक्रोप्रोसेसर में भी घटाया जाता है इसे उधार घटाना कहते हैं। दो डाटा में तुलना भी की जा सकती है।

उदाहरण

SUB CL, BL - CL= CL - BL

SUB DH, 6FH-DH = DH=6FH

SUBESI, 2000300H ESI = ESI-2000300H

SUB (DI), CH - DI द्वारा निर्देशित मेमोरी जगह के डाटा को CH से घटाना।

9.4 तुलना करना ओर बदलना (Comparison E Exchange)

80486, पेंटीयम तथा पेंटीयम प्रो प्रोसेसर में इन निर्देश का इस्तेमाल होता है ।

CMPXCHG- जगह के एड्रेस के डाटा की तुलना ACC के डाटा के साथ करना । इस निर्देश की दो अवस्था होती है पहली अवस्था में यदि दोनों सूची बराबर है तो स्त्रोत एड्रेस का डाटा (Oprand) नियत स्थान (Destination) में हो जायेगा । और यदि दोनों सूची बराबर नहीं है तो यह स्त्रोत ऐड्रेस का डाट ACC में कॉपी (Copy) होगा ।

पेटीयम तथा पेटीयम प्रो प्रोसेसर दो क्वाड शब्दों (Quad word) तुलना करता है तथा निम्न निर्देश का इस्तेमाल करता है।

(MP x CHG SB TEMP)

9.5 गुणा तथा भाग (Multiplication and Division)

80486, पेंटीयम तथा पेंटीयम प्रो प्रोसेसर गुणा तथा भाग दोनों है करते हैं।

9.5.1 गुणा (Multiplication)

इन माइक्रोप्रोसेसर में गुणा बाइट्स में शब्द में या दुगने शब्दों में कि जाता है । 80386 से परो पेंटीयम 32 बिट के दुगने शब्दों का गुणा करते हैं। यदि दो 8–बिट गुणित होती है तो 16-बिट का गुणनफल आता है और यदि 16-बिट का गुणा, 32 बिट का गुणनफल देता है।

गुणा करने के लिये यह ध्यान रखता अति आवश्यक होता है कि गुणा चिन्ह सहित पूणांक (Singed Integer) या चिन्ह रहित पूर्णांक (Unsigned inleger) हैं। इनके लिये निर्देश क्रमश: IMUL तथा MUL होते हैं। हमे यह विदित है गुणा करने के लिये एक गुण्य (Multiplicand) तथा एक गुणक (Multiplier) को होना अति आवश्यक होता है ।

गुण्य (Multiplicand) 8 बिट का AL रजिस्टर होता है तथा गुणक (Multiplier) कोई भी एक रजिस्टर या कोई भी मेमोरी जगह हो सकती है। गुणा निर्देश में डाटा के ऐड्रेस का होना अतिआवश्यक है क्योंकि यह AL रजिस्टर की सूची को Oprand बार गुणित करेगा। जैसे- MUL BL इसका मतलब यह हुआ कि AL की सूची की BL बार गुणित करता है।

IMUL DH, AL को DH के साथ गुणा करना तथा चिन्ह सहित गुणनफल को AX में रखना।

MUL TAM AL को मेमोरी जगह की इस बाइट से गुणित किया जाये जिसका ऐड्रेस TAM है तथा चिन्ह रहित गुणनफल को AX में रखे।

9.5.2 भाग (Division)

भाग को 8 बिट, 16 बिट तथा 32 बिट के साथ किया जा सकता है। भाग को भी चिन्ह सहित पूर्णांक तथा चिन्ह रहित पूर्णाक के द्वारा किया जा सकता है। इसके लिये क्रमशः IDIV तथा DIV निर्देश का इस्तेमाल किया जाता है। भाग के करने के लिये भाज्य (Dividend) तथा भागित सूची की आवश्यकता होती है। यह भाग किसी रजिस्टर में या किसी मेमोरी जगह में की जाती है।

9.6 बी.सी.डी. तथा आसकी अंकगणित (BCD and ASCII Arithmetic)

माइक्रोप्रोसेसर बाइनरी कोड डेसीमल (Binary Coded Decimal) तंत्र तथा (ASII-American Standard Code for) अंकगणितीय दस्तकारी (manipulation) का उपयोग करता है।

BCD अंक गणित में दो कार्य प्रणालियों का इस्तेमाल किया जाता है। वे भाग तथा घटाना है। माइक्रोप्रासेसर के निर्देश सेट ऐस दो निर्देश देता है जो BCD योग तथा घटाने के परिणाम को सही करता है। ये निर्देश निम्न हैं

1. DAA (Decimal adjust after addition)

2. DAS (decimal adjust after substraction)

डेसीमल को योग के पश्चात डीक करने के लिये ADD तथा ADC निर्देशों का इस्तेमाल किया जाता है। इसी तरह घटाने के पश्चात्य डेसीमल का सही करने के लिये किया विधि वही होती है। अब जोड़ने की जगह घटाने वाले निर्देशों का उपयोग किया जाता है ।

ASCII अंकगणित में यह फलन के साथ निर्देशित होता है और कोड अंक लिये होता है । इन अंकों की सीमा (Range) 30 H से 39H तक की होती है जिसमें 0-9 अंको का समावेश होता है। ASCII अंक गणित प्रचालन में चार निर्देशो का इस्तेमाल किया जाता है जो निम्न है।

AAA -ASCIT adjust after addition

AAD- ASCII adjust after Division

AAM- ASCII adjust after Multiplication

AAS- ASCII adjust after Substraction

AAA निर्देश (AAA Instruction)

Mov Ax, 32H; Load Ascit 2

ADDAL, 38H; Add Ascit 8

AAA ; Adjust

ADD Ax, 3030H Answer to Ascit

उपर्युक्त प्रोग्राम में पहले ASCII 2 को लोड किया गया है कि Ascit 8 के साथ जोड़ा गया है तथा AAA निर्देश देकर योग के बाद इसे ठीक गया है।

AAD निर्देश (AAD Instruction)

यह निर्देश हमेशा भाग देने से पहले दिया जाता है। इसमें रजिस्टर में बिना पैक किया हुआ 2 डिजिट का बी.सी.डी. नम्बर होता है AA रजिस्टर में AAD निर्देश के द्वारा ठीक करने के पश्चात्य इसे 2 डिजिट की बी. सी. डी. नम्बर से भागित किया जाता है जो एक एकल डिजिट के परिणाम को AL रखता है तथा जो शेष बचा है उसे AH में रखता है ।

AAM निर्देश (AAM Instruction)

एक डिजिट के दो बिना पैक के बी.सी.डी. नम्बरों को गुणित करने व बाद उन्हें AAM निर्देश द्वारा ठीक किया जाता है। AAM निर्देश का एक विशिष्ट उपयोग भी है यदि बाइनरी नम्बर 000H 00638 के मध्य है तो इसे निर्देश की सहायता से उन्हें बिना पैक किये बी. सी. डी. नम्बरों में बदला जा सकता है।

9.7 सामान्य लॉजिकल निर्देश(General Logical Instruction)

एक सामान्य लॉजिक में हम AND, OR EX-OR तथा NOT gate क इस्तेमाल करते हैं। 80486, पेटीयम तथा पेटीयम परो माइक्रोप्रसेसर में इन लोग प्रचलनों का उपयोग निम्न स्तरीय साफ्टवेयर में बाइनरी बिट को नियंत्रित करने लिये किया जाता है।

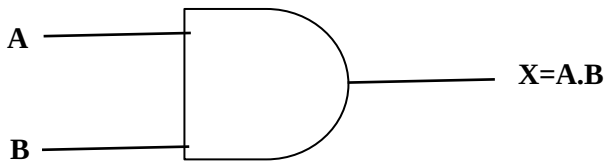
लॉजिकल निर्देश में TEST निर्देश जो AND निर्देश की एक विशिष नमूना होता है । इसीतरह NEG निर्देश होता है जो NoT निर्देश का एक नमूना हो है ।

जैसा कि हम जानते है कि यह बाइनरी बिट को Control करता है। यह निर्देश बिट को सेटए क्लयिर या काम्प्लीमेट करता है । यह फ्लेग बिट ऑफसेट करता है । यह हमेशा केरी को साफ करता है तथा फ्लेग को ओवर करता है जबकि अन्य दूसरे फ्लेग, परिणाम के विचार को ही बदल देते हैं ।

जब किसी बाइनरी डाटा को मेनीपुलेट किया जाता है दायी तरफ की पहली बिट को हमेशा रखा जाता है । इस तरह बिट की स्थिति 0 से बायी तरफ बिट 7 की ओर बढ़ती है यह क्रिया एक बाइट के लिये होती है और जब यह 15 की ओर बढ़ेगी तो क्रिया एक शब्द के लिये होगी ।

एक दुगने शब्द के लिये बिट स्थिति 31 का इस्तेमाल किया जाता है। जो सबसे बाई ओर कि बिट होती है।

9.7.1 AND प्रचालन (AND Operation)



A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

(चित्र- 7.1)

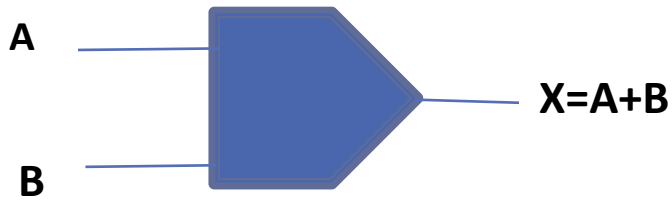
AND प्रचलन एक लॉजिक गुणा को प्रदर्शित करता है इसका लॉजिकल चित्र तथा सत्यतालिका चित्र (7.1) में दर्शायी गई है। इस प्रचालन में दोनों इनपुट उच्च होते हैं तभी आउटपुट उच्च आता है। इसके लिये AMD निर्देश का इस्तेमाल किया जाता है। यदि ज्यादा गति नही चाहिये तो इसे अलग (Discrete) AND गेट भी का जा

सकता है। सामान्यतः AND गेट को ROM Read Only Memory के साथ रखा जाता है जिससे इसका मूल्य काफी कम हो जाता है।

AND निर्देश को स्टोर करना (Store AND Instruction)

And निर्देश किसी बाइनरी नम्बर की तमाम बिट को साफ (Clear) कर सकता है। यह क्रिया मासकिंग (Masking) कहलाती है। सिवाय मेमोरी ऐड्रेस तथा रजिस्टर भाग एड्रेसिंग के AND निर्देश सभी ऐड्रेस मोड में इस्तेमाल किया जाता है।

8.7.2 OR प्रचालन (Or Operating)



A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

चित्र (7.2)

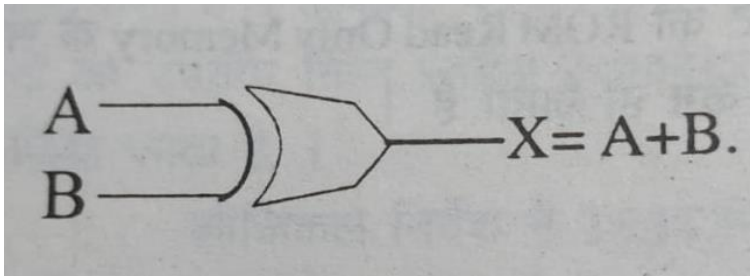
यह लॉजिकल जोड़ का प्रचालन करता है इसलिये इसे मिलाया OR फलन (Inclusive OR Function) भी कहते हैं। इसका सांकेतिक चित्र सत्य तालिका चित्र (7.2) में प्रदर्शित है।

OR प्रचालन की खासियत यह है कि इसे किसी के भी बदले के स OR किया जा सकता है। अगर OR गेट की चिप के साथ उपयोग किया जाये क महगा पड़ता है लेकिन यदि इसे ROM के साथ स्टोर किया जाये तो इसकी काफी कम हो जाती है।

OR गेट किसी भी बाइनरी नम्बर को सेट कर देता है अर्थात 1 पर करता है।

8.7.3 निवारक OR गेट (Exclusive OR gate)

इसको XOR गेट भी कहते हैं यह निवारक गेट इसलिये भी कहलाता है कि जब कोई भी एक इनपुट उच्च होता है तो आउट पुट उच्च होता है। इसका सांकेतिक चित्र तथा सत्य तालिका चित्र (73) में प्रदर्शित है।



A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

xxxx	xxxx	बिना जाना नंबर
0000	1111	मास्क
xxxx	xxxx	

9.8 परीक्षा तथा बिट परीक्षा निर्देश (Test and Bit Test Instruction)

परीक्षा निर्देश

(Test Inst परीक्षा निर्देश AND प्रचालन को करता है अन्तर सिर्फ इतना होता है कि AND निर्देश ओपरेंड (Operand) के नियत स्थान को बदल देता है जबकि परीक्षा निर्देश ऐसा नही करता है। यह केवल फ्लेग रजिस्टर को प्रभावित करता है। इसका फलन CMP की तरह होता है लेकिन Test केवल Multiple बिट पर प्रभाव डालता है जबकि CMP पूरी बाइट या शब्द पर प्रभाव डालता है ।

9.9 स्थान बदलना तथा घुमाना (Shift and Rotate)

जहाँ तक स्थान बदलने (shift) का प्रश्न है. यह दो प्रकार से किया जा सकता है। एक तो लॉजिकल स्थान बदलना (logically shift) तथा इसका अंकगणितीय स्थान बदलना (arithmetic shift) |

लोजिकल स्थान बदल दो प्रकार से किया जाता है। एक लोजिकल दोंयी तरफ स्थान बदल तथा लॉजिकल बाई तरफ स्थान बदल इसी तरह अंकगणितीय स्थान बदल भी दो प्रकार से किया जाता है।

1. अंकगणितीय दांयी तरफ स्थान बदल
2. अंकगणतीय बांयीं तरफ स्थान बदल

अंकगणितीय दांयी तरफ स्थान बदल में चिन्ह बिट को संख्या के साथ कॉपी (Copy) किया जाता है जबकि लॉजिकल दांयी तरफ स्थान बदल में नम्बर के माध्यम से 0 को कॉपी (Copy) किया जाता है। यहां ध्यान देने योग्य बात यह है कि दांयी तरफ स्थान बदल हमेशा दो से गुणित होता है जबकि बांयी तरफ स्थान बदल हमेशा दो से भागित होता है ।

10. माइक्रो कंट्रोलर

10.1 सेंट्रल प्रोसेसिंग यूनिट

एक कंप्यूटर में एक सीपीयू केंद्रीय रूप से निर्देश को प्राप्त करता है और संसाधित करता है यह प्रक्रिया निर्देश और अंकगणितीय और तार्किक संचालन, बिट हेरफेर और डेटा ट्रांसफर ऑपरेशन इनपुट है।

निर्देशों का सेट जिसे सीपीयू संसाधित कर सकता है, उसके विशिष्ट निर्देश सेट द्वारा परिभाषित किया गया है। निर्देश सेट प्रत्येक सीपीयू के लिए अद्वितीय है। सीपीयू में निम्नलिखित बुनियादी अनुभाग हैं।

(a) प्रोग्राम प्रवाह अनुभाग इसके कार्य

-कार्य मेमोरी या इनपुट आउटपुट उपकरणों से निर्देश और डेटा लाने के लिए हैं।

-किसी दिए गए आदेश या निर्देश के लिए प्रक्रिया को नियंत्रित करें और सिस्टम के साथ संवाद करें।

(b) निर्देश निष्पादन अनुभाग और सबसे महत्वपूर्ण इकाई,

-अंकगणितीय तर्क इकाई, एलयू प्रदर्शन कर सकते हैं।

-जोड़ या घटाव, गुणा, या विभाजन या तुलना। नोड ऑपरेशन या एक फॉर्म से दूसरे फॉर्म में रूपांतरण, या किसी गंतव्य में वृद्धि या ह्रास या स्थानांतरण।

-अन्य अंकगणितीय या तार्किक ऑपरेशन या बिट मैनिपुलेशन ऑपरेशन जैसे बाएं शिफ्ट, राइट शिफ्ट, रोटेट, लेफ्ट रोटेट लाइट डिजाइन और इसकी क्षमताओं के अनुसार।

चित्र(1) कंप्यूटर सिस्टम या एम्बेडेड सिस्टम में सीपीयू को मेमोरी इनपुट, आउटपुट डिवाइस और सिस्टम के भीतर निर्देश, डेटा और संचार लाने के लिए बस के साथ दिखाता है।

एक एम्बेडेड सिस्टम को एक कंप्यूटर सिस्टम के रूप में परिभाषित किया जा सकता है जिसमें सॉफ्टवेयर और ऑपरेटिंग सिस्टम एम्बेडेड होता है ताकि एक विशिष्ट उत्पाद या उसका एक हिस्सा प्रदान किया जा सके। एक बस समानांतर लाइनों का एक सेट है जो एक प्रणाली में दो सक्रिय खंड या उपखंड के बीच संकेत ले जाती है।

10.2 माइक्रो कंप्यूटर

चित्र(2) एक माइक्रो कंप्यूटर सर्किट की संरचना को दर्शाता है जैसा कि दिखाया गया है। एक माइक्रो कंप्यूटर एक सर्किट या एक वीएलएसआई कोर है जो कंप्यूटर के रूप में कार्य करता है और इसमें कम से कम निम्नलिखित शामिल होते हैं। वीएलएसआई कोर या एक एसआईसी स्पेस पिंग विशिष्ट एकीकृत सर्किट। कोर फ़ाइल को सेल लाइब्रेरी भी कहा जाता है। इसमें निम्नलिखित शामिल हैं।

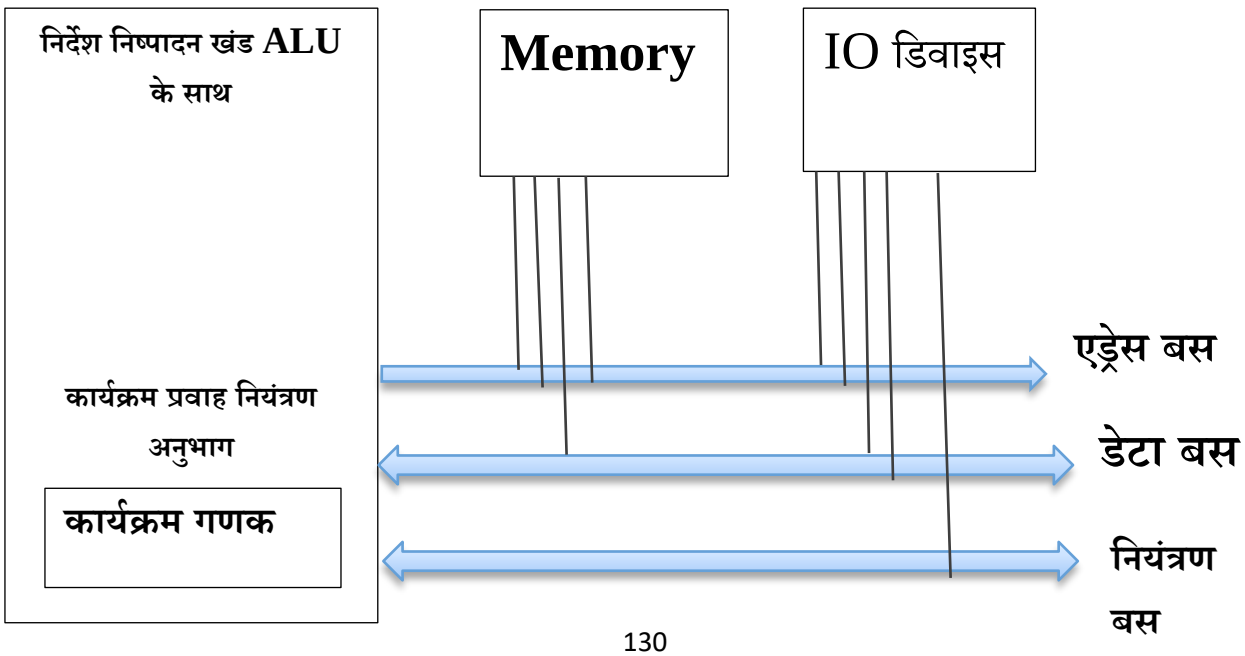
-माइक्रोप्रोसेसर।

-मेमोरी,

-डेटा और स्टिक के लिए मेमोरी।

-समर्पित कार्यक्रमों और उप कार्यक्रमों को संग्रहीत करने के लिए मेमोरी।

-सिस्टम को संभालने के लिए इंटरप्ट हैंडलिंग सर्किट या यूनिट इंटरप्ट।



10.3 कंप्यूटर सिस्टम

यह विशिष्ट कंप्यूटर एक विशेष सामान्य प्रोसेसर सिस्टम है जिसमें निम्नलिखित हार्डवेयर होता है।

-माइक्रो कंप्यूटर सर्किट या यूनिट?

-प्राथमिक और द्वितीयक के रूप में बड़ी स्मृति।

- इनपुट आउटपुट पोर्ट और डिवाइस जैसे सीओएम पोर्ट, कीबोर्ड पोर्ट और सिस्टम के लिए यूएसबी इनपुट आउटपुट डिवाइस जैसे कैमरा, डिजिटाइज़र और स्कैनर।

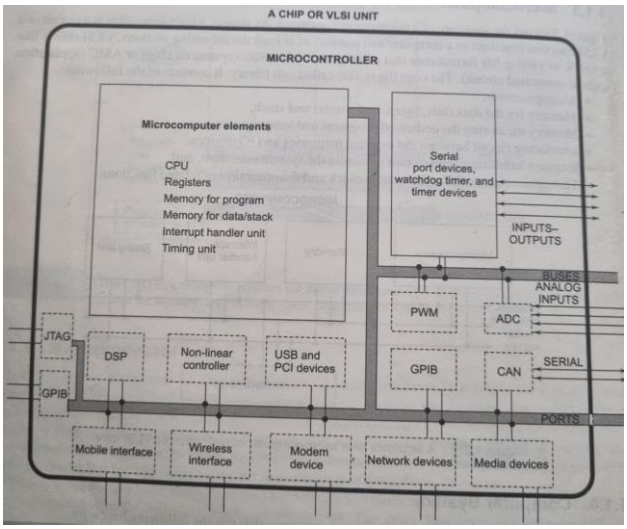
-आउटपुट पोर्ट और वीडियो मॉनिटर और प्रिंटर जैसे डिवाइस।

-इनपुट आउटपुट डिवाइस जैसे नेटवर्क, इंटरफ़ेस, यूनिट और मॉडेम।

10.4 माइक्रोकंट्रोलर ओर एम्बेडेड प्रोसेसर।

एक माइक्रोकंट्रोलर एक माइक्रो कंप्यूटर है जिसमें एकल चिप या वीएलएसआई कोर पर कुछ अन्य अनुप्रयोग विशिष्ट उपकरण होते हैं। यह एक वास्तविक समय नियंत्रण या संचार प्रणाली में एक एकीकृत हिस्सा है। इसमें निर्दिष्ट कम्प्यूटेशनल क्षमताएं और बढ़ी हुई इनपुट आउटपुट परिचालन क्षमताएं हैं।

टाइमिंगमाइक्रो कंट्रोलर यूनिट में एक माइक्रो कंप्यूटर सर्किट या यूनिट शामिल है जिसमें पावर डाउन मोड या निष्क्रिय मोड में चलने का प्रावधान है। इसमें निम्नलिखित उपकरणों में से कुछ या कई भी शामिल हैं।



बिट मैनिपुलेशन के साथ खराब डिवाइस, सीपीयू द्वारा निर्देश प्रसंस्करण जो इंटरफ़ेस, रिले और स्विच के नियंत्रण को सक्षम करता है।

- सिंक्रोनस और एसिंक्रोनस सीरियल 10 डिवाइस।

- सिस्टम घड़ी के लिए टाइमर डिवाइस, रियल टाइम क्लॉक सॉफ्टवेयर टाइमर, किसी घटना या सिग्नल का वास्तविक समय का पता लगाना, और वॉचडॉग टाइमर डिवाइस।

-पल्स चौड़ाई मॉड्यूलेशन।

-एनालॉग और डिजिटल रूपांतरण डिवाइस।

-मॉडेम डिवाइस.

- डीएसपी निर्देश प्रसंस्करण सर्किट 8 सीपीयू के साथ डिजिटल सिग्नल प्रोसेसिंग पोर्ट।

-आठ सीपीयू संसाधित करने वाले गैरेखिक नियंत्रक निर्देशों वाले पोर्ट।-

- नेटवर्क इंटरफ़ेस और नेटवर्क प्रसंस्करण से संबंधित सीपीयू प्रसंस्करण निर्देशों के साथ पोर्ट।

-मोबाइल या वायरलेस इंटरफ़ेस और सीपीयू के साथ खेल संबंधित निर्देशों को संसाधित करने में सक्षम है और

एम्बेडेड प्रोसेसर पिछले सटीक और गहन गणनाओं के लिए एक विशेष माइक्रोप्रोसेसर है और जटिल वास्तविक समय अनुप्रयोगों के लिए यह एक प्रोग्राम से दूसरे में त्वरित स्विचिंग प्रदान करता है। साझा डेटा से जुड़ी समस्याओं को हल करने के लिए और एम्बेडेड सॉफ़्टवेयर द्वारा तेज़, अधिक सटीक और गहन गणना के लिए आरएससी कोर का उपयोग करें।

10.5 माइक्रोकंट्रोलर के प्रकार

चित्रा आंतरिक बस चौड़ाई, एम्बेडेड माइक्रोकंट्रोलर, निर्देश सेट, मेमोरी आर्किटेक्चर, आईसी चिप के संदर्भ में वर्गीकृत विभिन्न प्रकार के माइक्रोकंट्रोलर को दर्शाता है।एक ही परिवार के लिए वे विभिन्न स्रोत हो सकते हैं, उन्हें दिखाया गया है।

8 ,16 और 32 बिट माइक्रोकंट्रोलर

10.5.1 **8 बिट माइक्रोकंट्रोलर**

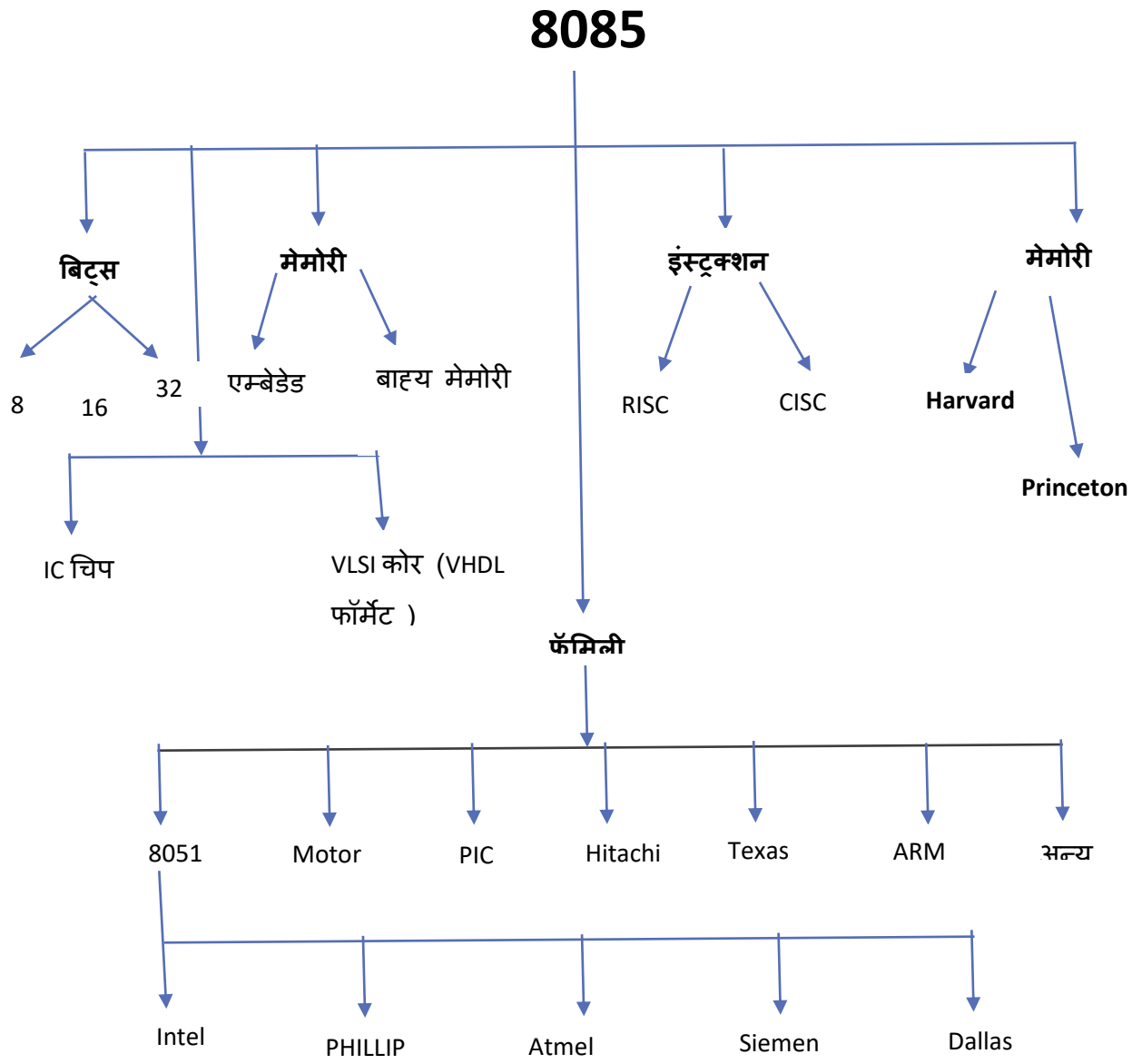
जब आंतरिक बस एक एमसीयू है तो 8 बिट है और एएलयू एक निर्देश पर काटने पर अंकगणित और तर्क संचालन करता है, एमसीयू एक 8 बिट माइक्रोकंट्रोलर है। कुछ 8 बिट माइक्रोकंट्रोलर इकाइयाँ इंटेल 8031/8051 परिवार और मोटोरोला MC68HC11 हैं।

10.5.2 16 बिट माइक्रोकंट्रोलर

जब आंतरिक बस एक एमसीयू है तो 16 बिट है और एएलयू एमसीयू के निर्देश पर 16 बिट्स के शब्द पर अंकगणित और तर्क संचालन करता है। महत्वपूर्ण 16 बिट मैकस विस्तारित 8051एक्स, एक इंटेल 8096 परिवार और मोटोरोला एमसी 68एससी हैं। ये 8 बिट माइक्रोकंट्रोलर इकाइयों की तुलना में अधिक परिशुद्धता और प्रदर्शन प्रदान करते हैं।

10.5.3 32 बिट माइक्रोकंट्रोलर

जब माइक्रोकंट्रोलर इकाई में डेटा ट्रांसफर ऑपरेशन के लिए आंतरिक बस 32 बिट्स होती है और एएलयू 32 के शब्द पर अंकगणित और तर्क ऑपरेशन करता ,फिर माइक्रोकंट्रोलर इकाई एक 16 बिट माइक्रोकंट्रोलर है। महत्वपूर्ण 32 बिट माइक्रोकंट्रोलर इकाइयाँ इंटेल तिरछी एटमेल 251 परिवार हैं। इंटेल 80960 परिवार, मोटोरोला एम 683 एक्स परिवार और एआरएम 7 या 9 या 11 प्रोसेसर आधारित सिस्टम। ये 16 बिट माइक्रोकंट्रोलर इकाइयों की तुलना में अधिक परिशुद्धता और प्रदर्शन प्रदान करते हैं और मोबाइल फोन जैसे एप्लिकेशन के लिए एम्बेडेड कंप्यूटिंग सिस्टम में एप्लिकेशन ढूंढते हैं। छवि प्रसंस्करण आधारित उत्पादों और एयरोस्पेस प्रणाली।



10.6 एम्बेडेड और बाहरी मेमोरी माइक्रोकंट्रोलर

10.6.1 एम्बेडेड मेमोरी माइक्रोकंट्रोलर

जब एक एम्बेडेड सिस्टम में एक माइक्रोकंट्रोलर यूनिट होती है जिसमें एक इकाई में सभी हार्डवेयर और सॉफ्टवेयर यूनिट होते हैं, तो माइक्रोकंट्रोलर यूनिट को एम्बेडेड माइक्रोकंट्रोलर कहा जाता है। नियंत्रण के दौरान या बाहरी उपकरणों का उपयोग करते समय प्रसंस्करण के लिए बहुत कम या कोई अन्य बाहरी इकाई या सिस्टम मौजूद नहीं है।उदाहरण के लिए, एक मोबाइल फोन में एक एम्बेडेड माइक्रोकंट्रोलर होता है।

10.6.2 बाहरी मेमोरी माइक्रो नियंत्रक।

जब एक एम्बेडेड सिस्टम में एक माइक्रोकंट्रोलर यूनिट होती है जिसमें सभी हार्डवेयर और सॉफ्टवेयर यूनिट मौजूद होते हैं, न कि एक इकाई के रूप में, और बाहरी रूप से मेमोरी इकाइयों का सभी या हिस्सा होता है। गोंद सर्किट कहा जाता है। एमसीयू को बाहरी मेमोरी माइक्रोकंट्रोलर कहा जाता है। उदाहरण के लिए, 8031 में 8051 के विपरीत प्रोग्राम मेमोरी बाहर है।

10.7 सीआईएससी एंड आर आईएससी आर्किटेक्चर माइक्रोकंट्रोलर

जटिल निर्देश सेट, कंप्यूटर आर्किटेक्चर, माइक्रोकंट्रोलर

जब एक एमसीयू में एक निर्देश सेट होता है जो अंकगणित और तार्किक निर्देश के लिए और मेमोरी आकलन और डेटा ट्रांसफर निर्देश के लिए कई एड्रेसिंग मोड का समर्थन करता है।एमसीयू को सीआईसी आर्किटेक्चर का बताया जाता है सीआईसी डेटा ट्रांसफर और अंकगणित और अन्य ऑपरेशन करने के विभिन्न तरीकों को चुनने में लचीलापन प्रदान करता है। उदाहरण के लिए, सीआईएससी में दो रजिस्टर या एक रजिस्टर और एक मेमोरी या दो मेमोरी पते की सामग्री जोड़ना संभव है?

निर्देश सेट कंप्यूटर माइक्रोकंट्रोलर को कम करें

जब एक एमसीयू में एक अनुदेश सेट होता है जो अंकगणितीय और तार्किक निर्देशों के लिए एक या दो एड्रेसिंग मोड और डेटा ट्रांसफर निर्देश के लिए कुछ का समर्थन करता है।एमसीयू को आरआईएससी आर्किटेक्चर का बताया जाता है।इन दिनों, अधिकांश माइक्रोप्रोसेसर और माइक्रोकंट्रोलर डिज़ाइन आरआईएससी कोर पर आधारित हैं, क्योंकि सीआईएससी सुविधाओं को हमेशा उपयुक्त प्रोग्रामिंग के लिए प्रदान किया जा सकता है।आरएससी कोड के लिए कोड का अनुवाद करने के लिए।

हार्वर्ड और प्रिंसटन मेमोरी आर्किटेक्चर माइक्रोकंट्रोलर

जब एक एमसीयू में प्रोग्राम मेमोरी और डेटा मेमोरी के लिए एक अलग मेमोरी एड्रेस स्पेस होता है, तो एमसीयू में हार्डवेयर हेवर्ड मेमोरी आर्किटेक्चर होता है।इसमें

अलगअलग निर्देश हैं और इसलिए- इन दो यादों से डेटा हस्तांतरण के लिए अलग-अलग नियंत्रण संकेत हैं।

जब एक एमसीयू में एक सामान्य मेमोरी एड्रेस स्पेस होता है, तो उपयोगकर्ता प्रोग्राम मेमोरी और डेटा मेमोरी के लिए होगा। एमसीयू में प्रोसेसर में प्रिंसटन मेमोरी आर्किटेक्चर है।

मौलिकता/अनापत्ति प्रमाण-पत्र

यह प्रमाणित किया जाता है कि मेरे/हमारे द्वारा लिखित पुस्तक माइक्रोप्रोसेसर एवं माइक्रोकंट्रोलर [एक परिचय]
(पुस्तक का नाम) मेरी/हमारी मौलिक एवं अप्रकाशित कृति है। मैं/हम इसके मुद्रण/प्रकाशन की सहर्ष अनुमति देता हूँ/देते हैं।

यदि डॉक्टर हरीसिंह गौर विश्वविद्यालय, सागर/विश्वविद्यालय अनुदान आयोग/शिक्षा मंत्रालय, भारत सरकार इस पुस्तक को मुद्रित/प्रकाशित करवाता है तो मुझे/हमें कोई आपत्ति नहीं होगी।

स्थान : सागर
दिनांक : 15/07/23

लेखक के हस्ताक्षर एवं नाम : .....
Prof. Ashish Verma

सह-लेखक के हस्ताक्षर एवं नाम : .....
Ritu Arya