المعالج الدقيق أو (microprocessor).

وهو عبارة عن شريحة إلكترونية دقيقة الصنع وصغيرة الحجم تقوم بكافة عمليات المعالجة. بمعنى أن وحدة المعالجة المركزية CPU في الحاسوب الشخصي هي عبارة عن المعالج الدقيق Microprocessor. هو العقل المدبر للحاسب ، يستقبل الاوامر ويعالجها ويعطينا نتائجها على شكل معلومات نستفيد منها ، من الناحية الاعتيادية هو قطعة مربعة الشكل وخفيفة الوزن يخرج من أسفلها عدد من الابر (pins) التي تسمح للمعالج بالاتصال مع مقبس المعالج على اللوحة الام وذلك لتبادل البيانات بينه وبين اللوحة الأم ، يتكون في الاصل من ملايين الترانزستورات المجموعة في شريحة صغيرة جدا من السليكون ، وهذه الشريحة تثبت من قبل المصنّع للمعالج على غلاف المعالج (القطعة المربعة) او داخلها وذلك لايصالها بالابر التي تكون أسفل غلاف المعالج .

- المهام الاساسيه للمعالج:

1. يجب ان يكون المعالج قادرا على احضار معلومات من الذاكرة
2. يجب ان يحتوي المعالج على مكان مناسب بداخله لحفظ هذهالمعلومات التي احضرها لحين الحاجه اليها.
3. لابد ان تكون لديه الوسائل المناسبه لادخال معلومات من بوابة الادخال حتى يتسنى لنا قراءة لوحة المفاتيح او ادخال دراجة حرارة
4. يجب ان يكون لدية المقدرة على اجراء العمليات الحسابية والمنطقية.
5. المقدرة على ارسال بيانات الى الذاكرة وتسجيلها فيها
6. المقدرة على ارسال بيانات الى وحدات الاخراج

- وظيفه المعالج:

المعالج يعتبر من أسرع وحدات الحاسب حيث يمكنه تنفيذ عدة ملايين من العمليات الرياضية فى الثانية الواحدة ، نظراً لسرعة الحاسب فإنه يحتاج بجواره إلى مخزن للأوامر المطوب منه تنفيذها حتى لا ينتظر وصول هذه الأوامر من الذاكرة الإلكترونية ولذلك تم تصميم شريحة إلكترونية تسمى ذاكرة الكاش Cache لكى تمد المعالج بسرعة هائلة بالأوامر التى سيقوم بتنفيذها ، كلما كانت ذاكرة الكاش Cache حجمها كبير كلما تمكن المعالج من الحصول على الأوامر التى يطلب منه تنفيذها بسرعة وبالتالى تزداد سرعة الحاسب بوجه عام .

- تتألف المعالجات من عدد كبير جداً من الترانزسترات ، فما هو عمل هذه الترانزسترات ؟ ومما يتكون ؟

إن المعالج يقوم مبدأ عمله على التعامل مع البيانات على شكل بتات (Bits) وبايتات(Bytes)، فالمعالج لا يفهم إلا لغة البايتات(Bytes) على شكل واحدات (Ones) و أصفار (Zeros)، بالنسبة للمستخدم فإن البايتات قد تعني في نهاية المطاف صورة أو رسالة أو ...أو... أما بالنسبة

للمعالج فهي واحدات وأصفار .. كل بت يعتبره شحنة ويتعامل معه على أنه شحنة ينقلها ويخزنها هكذا .

وإذا نظرنا نظرة متعمقة في داخل المعالج ونظرنا لما يعمله المعالج نجد أنه **إما** يقوم بالعمليات الحسابية كالجمع والطرح ..إلخ **أو** يقوم بالعمليات المنطقية كالمقارنة بين الأعداد ، وفي كل الأحوال على المعالج أن يتخذ - بمساعدة التعليمات - القرارات الصحيحة ويقود دفة العمل على هذا الأساس .

- فكيف يتخذ الحاسب القرارات ؟

إن هذا هو عمل الترانزسترات (Transistors) ، ولا تحسب أن ترنزستر واحد يستطيع أن يقوم باتخاذ القرارت بل إن هذه الترانزسترات(Transistors) موزعة في شكل مجموعات داخل المعالج لتقوم كل مجموعة منها بنوعية معينة من الأعمال ، فمثلاً أحد المجموعات مخصصة للمقارنة بين الأرقام و أخرى لاتخاذ القرارات في حالة معينة وهكذا ، وفي كل مجموعة تختلف عدد وطريقة تجمع الترانزسترات مما يؤثر على وظيفتها ، ويستطيع الحاسب باستخدام هذه المجموعات المختلفة بشكل مدروس ومنظم أن يقوم بكل العمل الذي يطلب منه .

إن كل "مجموعة" من هذه المجموعات تسمى "Logic Gates – بوابة منطقية" وتختلف البوابات المنطقية بحسب الوظيفة التي تؤديها وعدد الترانزسترات(Transistors) التي تحتويها.

وتصنيع المعالج ماهو إلا وضع هذه المجموعات وربطها ببعضها بالشكل المطلوب ، إن "المجموعات" إذا تجمع عدد كبير منها لأداء وظيفة معينة تصبح ما نسميه IC والمعالج ما هو إلا مجموعة من الـ IC مترابطة مع بعضها البعض بشكل

تنقسم المعالجات من حيث عدد البتات الى:

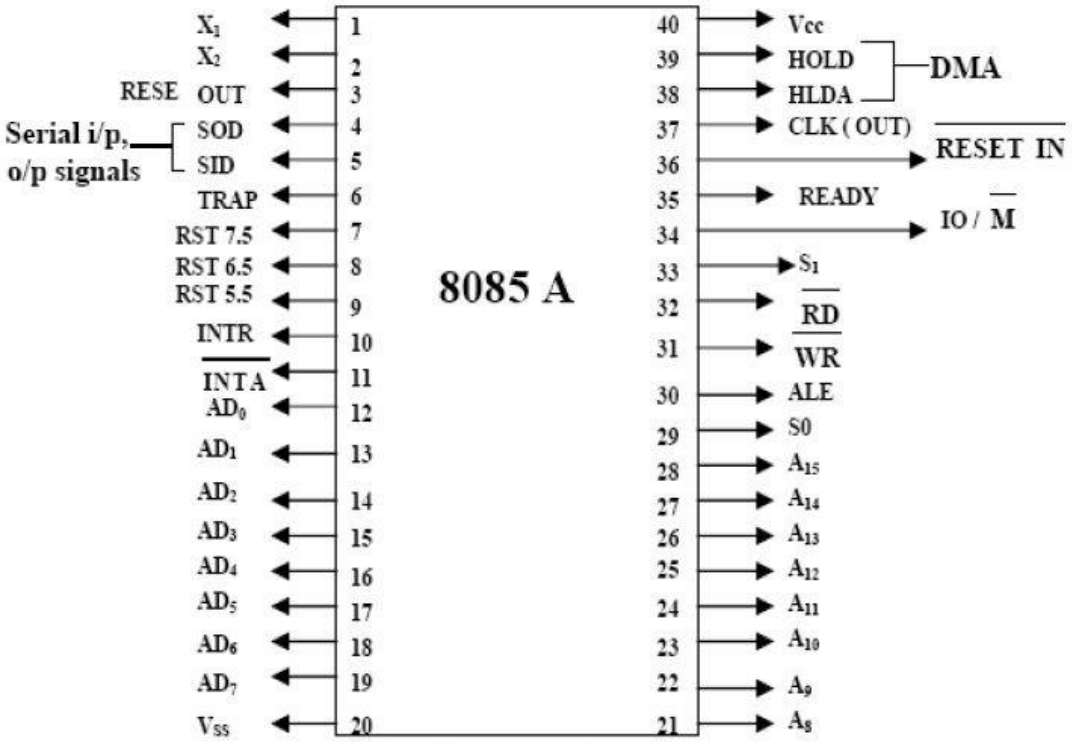
1. معالجات ذات ال 4- Bits : مثل Intel 4004 وكان اول معالج يستخدم.
2. معالجات ذات ال8-Bits : 8008, 8080, 8085
3. معالجات ذات ال 16-Bits :مثل 8088, 8086
4. معالجات ذات ال32-Bits :مثل Intel 386, Intel 486 , AMD486
5. معالجات ذات ال64-Bits :مثل Pentium, Pentium2, Pentium3 , Pentium4

سندقق النظر في هاذ الفصل الدراسي على تركيب شريحتين من شرائح الجيل الثاني من المعالجات وهي شرائح Intel 8085 و Z80 بالاضافة الى شريحة الجيل الثالث وهي Intel 8086 و ذلك لانها الاسهل في التعليم و الابسط في التركيب و الانسب لتقديم فكرة المعالج و كيفية عمله و برمجته للمتعلمين الجدد في هذا المجال.

المعالج الرقمي Intel8085 :

المعالج الرقمي 8085 هو احد معالجات شركة انتل للجيل الثاني ذو ال 8-Bits يحتوي على ذاكره عشوائيه RAM وعلى 7 Registers يتم برمجتهُ بلغة التجميع او ما تُسمى الأسمبلي وتعتبر من اللغات ذات المستوى الواطىء أذ أنها لاتحتوي على أيعازات متقدمه مثل الضرب او القسمه او الشرط ولكن يتم بناء الأيعاز أذ أن الضرب هو عملية جمع متكرر والقسمه هي عملية طرح متكرر. المعالج 8085 يعتبر من المعالجات القديمه يمكن من خلاله تصميم معادلات رياضيه ,اشارات تحكم,مصابيح الخ..

الرسم التخطيطي لاطراف المعالج 8085



مخطط الاطراف الخارجيه للمعالج الرقمي 8085 Pin Diagram

يتكون المعالج الدقيق 8085 من 40 PIN موزعه كما في الشكل اعلاه . وفي ما يلي بيان بالتعريفات الوظيفية لهذه الاطراف :

1. **X1 & X2 (مدخل) :**

يستخدمان لتوصيل مولد نبضات توقيت خارجية و ذلك لتحديد تردد نبضات الساعة الداخلية او توصيل للبلورة .

2. **RESET OUT (مخرج) :**

تبين انه حدث تصفير للمعالج .عندما يكون عاليا فانه يشير الى ان CPU قد تمت اعادة تهيئته هذا يعني انه قد تمت اعادة تهيئة عداد البرنامج و سجل الايعاز...الخ الى الصفر تذهب هذه الاشاره الى الرقاقات المحيطه . عندما يوصل مصدر القدرة للوهلة الاولى تعاد تهيئة المنظومه باكملها بما في ذلك المعالج الدقيق . بعد ان تصبح اشارة ال RESET OUT واطئة تبدأ عملية المعالجة .

3. **SOD (مخرج) :**

و تعني بيانات اخراج متواليه (Serial Out Data) حيث يوجد برنامج يحول بيانات المركم الى سيل من بيانات متوالية هذه البيانات المتوالية تخرج من هاذ الطرف و الذي يمكن ربطه الى وسيلة اخراج متوالية.

4. **SID(مدخل) :**

و يعني بيانات ادخال متوالية (Serial In Data) حيث ان هذا الطرف طرف ادخال للبيانات المتوالية . و توجد ايعازات تحول بيانات الادخال المتواليه الى صيغه ال 8-bits .

5. **الاطراف من 6 الى 11 :**

هذه الاطراف هي جزء من وحدة سيطرة المقاطعه . حيث ان المعالج 8085 له خمسة ادخالات لمتطلبات المقاطعه . توجد اسبقية لاطراف المقاطعه حيث ان لبعضها اهميه اكثر من البعض الاخر . يكون ترتيب اشارات المقاطعه الخمس حسب اهميتها كما يلي
INTR, RST5.5 , RST6.5 , RST7.5 , TRAP . اذا اصبح اثنان او اكثر من المقاطعات عاليا في نفس الوقت فان المعالج سوف يستجيب لها حسب اهميتها (TRAP اولا وهكذا)

- TRAP(مدخل) : بداية المقاطعه و لايمكن منعها .
- RST5.5, RST6.5, RST7.5(مدخل): اعادة البدء بالمقاطعة .
- INTR(مدخل) : طلب مقاطعة ويستعمل كمقاطعة عمومية ويمكن السماح او عدم السماح بالمقاطعة .
- INTA(مخرج) : الموافقة على المقاطعة و ذلك لادخال تعليمات اعادة بدء او تعليمات استدعاء .

6. **الاطراف من 12 الى 19**

AD0……AD7 (مداخل او مخارج) : خطوط نقل ثنائية الاستخدام حيث تستخدم لنقل العناوين والبيانات , تكون موزعه بهذا لاشكل من اجل الحفاظ على عدد اطراف المعالج 40 طرف.

7. **A8……A15 (مخارج) :**

خطوط العناوين والتي تحمل الاشارات الثمان ذات القيمة الاعلى حيث تكون الاشارات الثمانية الاخرى على الخطوط AD0…….AD7 .

8. **S0,S1 (مخرج) :**

تمثل هذه المخارج اشارات تحكم والتي تقوم باخطار الوحدات الاخرى بنوع العمل الذي يقوم به المعالج الدقيق اذا كان قراءة او كتابة او جلب ايعاز حسب الجدول التالي:

الحالة	S0	S1
HALT	0	0
WRITE	1	0
READ	0	1
FETCH	1	1

9. **ALE(مخرج) :**

و تعني هذه الاشاره تمكين قفل العنوان (Address Latch Enable)وهي اشارة تستخدم لبيان ان اشارات العنوان موجوده على خطوط العناوين والبيانات ليتم تخزينها . حيث اذا كانت ALE=1 فان الطراف من 12 الى 19 تكون خطوط عناوين وبعكس ذلك تكون خطوط بيانات .

10. **الاطراف 31 و 32 و 34 :**

تعمل هذه الاطراف الثلاثه مجتمعه

- IO/M(مخرج) : يبين ما اذا كانت عملية القراءة او الكتابة الى الذاكره او الى جهاز الادخال او الاخراج . حيث تشير IO/M واطئه الى عملية قراءة او كتابة من الذاكرة

, واذا كانت هذه الاارة عاليه فان عملية الكتابة او القراءة تكون من جهاز اخال او اخراج .
الاشارتان التاليتان تحددان فيما اذا كان هناك عملية قراءة او كتابة حيث ان هاتين الاشارتين واطئتا الفعاليه ولا يمكن ان تعملان في نفس الوقت .

- WR(مخرج) : اشارة كتابة تبين ان البيانات موجودة على خطوط البيانات وسيتم كتابتها في مكان من الذاكرة او جهاز الادخال او الاخراج .
- RD(مخرج) :اشارة قراءة تبين ان محتويات الذاكرة او جهاز الادخال او الاخراج سيتم قراءتها وان خطوط البيانات جاهزة لنقل البيانات .

كمثال : خلال عملية قراءة من الذاكرة فان اشارة IO/M يجب ان تكون واطئه اما اشارة WR فيجب ان تكون عالية و اشارة RD يجب ان تكون واطئه .

11. **READY (مدخل) :**

جاهز وهي اشارة تدخل الى المعالج الدقيق لاخطاره بان الوحدات الاخرى جاهزه لاستقبال او ارسال بيانات .

الفكرة هي : يعنون المعالج الدقيق 8085 الوسيلة المحيطه كخطوة اولى في ارسال او استقبال بيانات من تلك الوسيلة . اذا كانت الوسيلة غير مهيأة فانها سترجع اشارة READY واطئة الى المعالج حيث يكون المعالج هنا في حالة انتظار WAIT . وعندما تكون الوسيلة جاهزة فانها ترسل اشارة READY عالية الى المعالج وبذلك يكون المعالج قادرا على اكمال تحويل البيانات .

12. **RESET IN (مدخل) :**

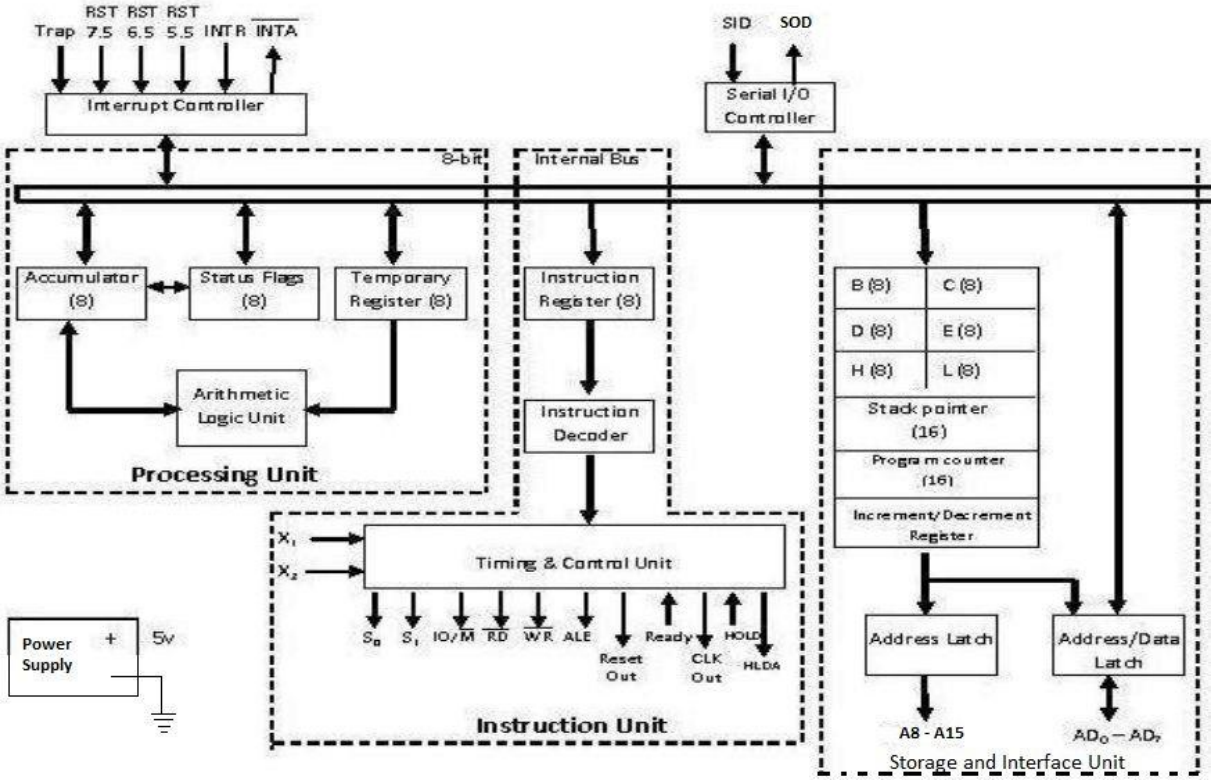
اشارة تقوم باعدة ضبط (تصفير) وتجعل عداد البرنامج يساوي صفرا .

13. **CLK(OUT) (مخرج) :**

تخرج من الطرف 37 و هي مؤقت للمنظومة حيث تذهب اشارة ال CLK الى الرقاقات المحيطه و تزامن توقيتاتها.

14. **HLDA (مخرج) :** الموافقة على طلب الامساك .
15. **HOLD (مدخل) :** اشارة تقوم باخطار المعالج الدقيق بان جهاز اخر يريد استخدام خطوط العناوين والبيانات .
16. **VCC :** مصدر كهربائي ذو جهد +5V .
17. **VSS :** طرف ارضي .

المخطط الكتلى للمعالج 8085



الشكل يعطي نظره شاملة على مكونات المعالج 8085

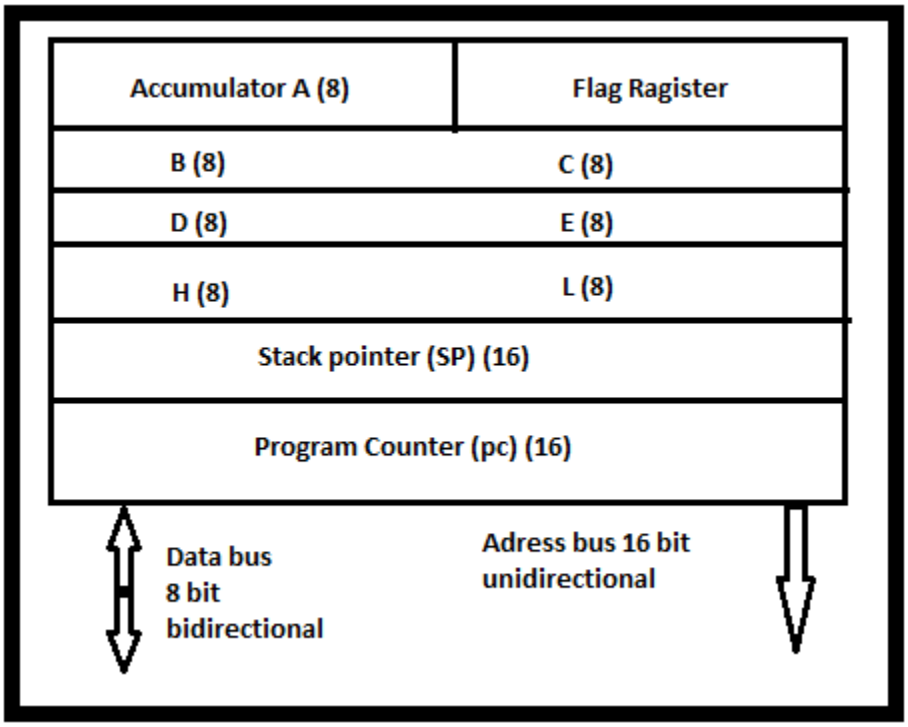
جميع المعالجات تتكون من الاجزاء التاليه :

1. المسجلات والعدادات (Register section) .
2. وحدة الحساب والمنطق (Arithmetic Logic Unite) .
3. وحدة التحكم والتزامن (Timing and Control Unite) .

المسجلات والعدادات (Register section) :

تستخدم المسجلات للتخزين الؤقت للمعلومات في صورة خانات ثنائيه في داخل شريحة المعالج لحين الحاجه اليها , والمسجلات في المعالج هي واحد من نوعين:

1. مسجلات عامه الاغراض General Purpose Register : و تؤدي اكثر من وظيفة للتسجيل فيها او القراءة منها
2. مسجلات خاصة الاغراض Dedicated Register :وتودي وظيفة واحدة فقط .



شكل يوضح المسجلات في المعالج الرقمي 8085

اما العدادات Counters فستخدم لعد النبضات الداخلة اليها تصاعديا او تنازليا

فيما يلي نتعرف على وظيفة كل مسجل وعداد في المعالج :

1. مسجل المركم A Accumulator

يعتبر مسجل المركم اكثر مسجلات المعالج عملا واي عملية حسابية او منطقية يقوم بها المعالج يجب ان يكون مسجل المركم طرفا فيها, بالاضافة الى ان اي نتيجة حسابية او منطقية لا توضع الا في مسجل المركم . وايضا اي عملية ادخال او اخراج يقوم بها المعالج من بوابات الادخال او الاحراج تكون عادة من خلال مسجل المركم . وعدد الخانات في مسجل المركم يساوي عدد خطوط البيانات وهو 8-Bits .

2. مسجلات عامه الاغراض General Purpose Register

ان المسجلات العامه هي B,C,D,E,H,L ستة مسجلات يمكن ان تستخدم كمسجلات ذات ال 8-Bits كل مسجل على حدة ويمكن ان تستخدم كمسجلات ذات ال 16-Bits معتمدا على طول الامر المراد تنفيذه مثل المسجل المزدوج BC او DE او HL . كمعظم الحاسبات الدقيقة فان المسجل المزدوج HL يدعى موشر البيانات ويمكن ايضا استخدامه كموشر للعناوين وتوجد هناك بعض الاموامر البسيطة والتي تستخدم المسجل المزدوج BC,DE كموشرات للعناوين ولكن عادة يستخدم هذان المسجلان المزدوجان كمسجلين للبيانات .

ضمن هذه المسجلات يوجد ايضا مسجل المكدس و عداد البرنامج و (المزيد – المنقص Increment-Decrement) حيث يمكن للمزيد – المنقص من ان يزيد او ينقص من محتويات موشر المكدس او عداد البرنامج .

3. عداد البرنامج Program Counter (PC)

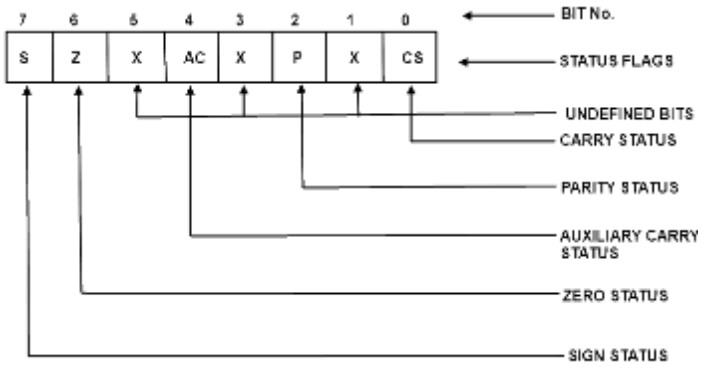
عداد ذو 16-Bits يستخدم كموشر الى عنوان موقع الذاكرة للامر التالي المراد تنفيذه

4. مسجل موشر المكدسة Stack Pointer Register (SP)

يعتبر جزء من الذاكرة يتم فيه تخزين بعض العناوين او البيانات المهمه والتي لابد من الحاجه اليها واسترجاعها مرة ثانية وبنفس الترتيب الذي تم تخزينها به وهو من المسجلات ذو ال -16 Bits .

5. مسجل الحالة Status Register (SR)

ويطلق عليه اسم مسجل الاعلام Flag Register يعكس هاذ المسجل حالة نتيجة اخر عملية حسابية او منطقية قام المعالج بتنفيذها حيث ان هذا المسجل يكون ذو 8-Bits كل بت يمثل حالة معينة يمثل كما في الشكل ادناه:



الشكل 3-2 مسجل الحاله في المعالج 8085

- **Sign Flag(s)**: هذا البت يعطي اشارة نتيجة اخر عملية حسابيه او منطقية قام بها المعالج فإذا كانت SF=0 فالعدد موجب أما إذا كانت SF=1 فالعدد سالب.

- **Zero Flag (ZF)**: ياخذ هاذ البت قيمة واحد اذا كانت نتيجة اخر عملية حسابيه او منطقية تساوي صفر اما في ماعدا ذلك تكون قيمة هاذ البت صفر.

- **Auxiliary Carry Flag (AC)**: تكون قيمة هاذ البت واحد اذا كان هناك حمل من الخانة الثالثة الى الخانة الرابعه نتيجة اي عملية جمع او هناك استلاف من البت الرابع الى البت الثالث نتيجة اي عملية طرح.
- **Parity Flag (PF):** اذا كان نتيجة اخر عملية حسابية او منطقية تحتوي على عدد زوجي من الواحدات فيكون هاذ البت يساوي واحد اما اذا كان عدد الواحدات فردي فان هاذ البت تكون قيمته صفر.

- **Carry Flag (CY):** تكون قيمية هاذ البت واحد اذا حصل حمل Carryمن اخر بت في اي عملية حسابية او حصل استلاف Barrowمن اخر بت في اي عملية طرح.

6. مسجل الامر و فاك الشفرة Instruction Register And Decoder

يحتوي مسجل الامر على شفرة الامر الذي يتم تنفيذه الان وهو ذو ال 8-Bits .وفي المقابل تنتقل محتويات الامر الى فاك الشفرة Decoder وبالتالي فان نتيجة فاك الشفرة ينتقل على شكل نبضات تتحكم في عملية تنفيذ البرنامج لكل من المسجلات و وحدة الحساب والمنطق .

المسجل Register	الشفرة Code
A	111
B	000
C	001
D	010
E	011
H	100
L	101
M	110
ازواج المسجلات Register Pairs	
BC	00
DE	01
HL	10
SP	11

المسجلات وازواج المسجلات وشفراتها الثنائيه

وحدة الحساب والمنطق Arithmetic and Logic Unit

ALUهي اختصار لعبارة Arithmetic and Logic Unit وهي دارة رقمية موجودة داخل وحدة المعالجة المركزية مسؤولة عن اجراء كافة العمليات الحسابياتية (كالجمع والطرح والقسمة الضرب) علما ان الوحدة فقط تستخدم عملية واحدة وهي الجمع اما العمليات الأخرى مثل الطرح هي عملية جمع المكافئ والضرب هي جمع لعدة مرات والقسمة هي والعمليات المنطقية (مثل OR,AND,XOR) في الحواسب، كما تقوم بعمليات المقارنة لمعرفة نتيجة المقارنات المنطقية وهي : (أكبر من وأصغر من ويساوي ولا يساوي) ومشتقات هذه المقارنات، وكذلك فهي توفر إمكانية تخزين المعلومات بشكل مؤقت بالإضافة إلى إمكانية معالجة المعلومات. فهي تعطي خرجها بالاعتماد على قرار متخذ بداخلها. وبما أن كافة عمليات المعالجة تنحصر في نوعين من العمليات فإما أن تكون حسابياتية أو ان تكون منطقية

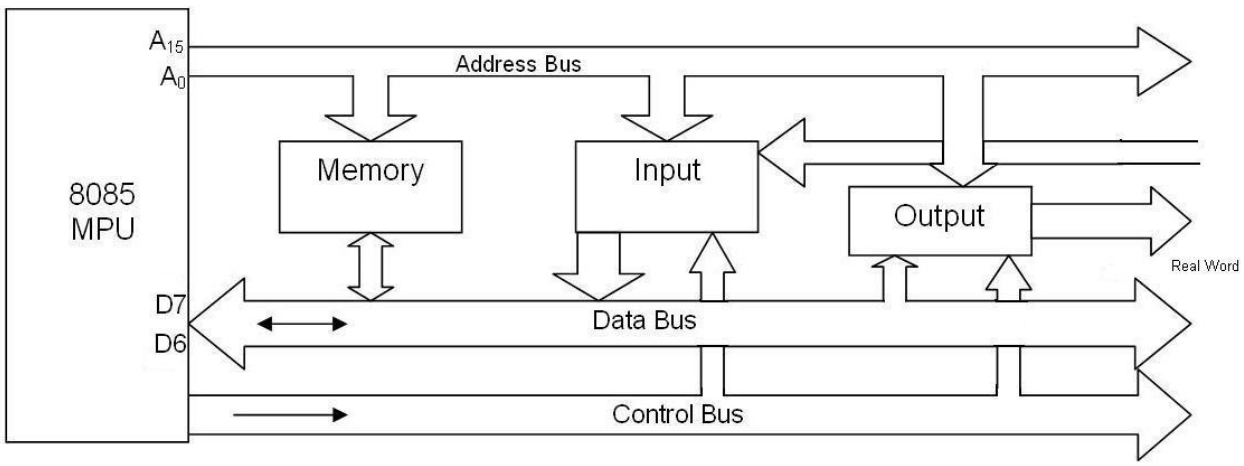
فإن هذه الوحدة قادرة على معالجة أي مسألة يطلب منها معالجتها، ويمكن القول ان هذه الوحدة هي التي تقوم فعليا بتنفيذ التعليمات.

وحدة التحكم والتزامن Timing and Control Unit :

تقوم وحدة التحكم بالتحكم في عمل الدوائر المختلفة حيث انها تستطيع تحديد الدوائر التي يجب ان تعمل لتنفيذ عمل ما وتحافظ على عملية التزامن لعمل هذه الدوائر . تقوم هذه الوحدة بارسار اشارات التحكم الضرورية لتنفيذ الامر الى الدوائر او الاجزاء المختلفه لكي تعمل على تنفيذ الامر وهي متصلة بجميع اجزاء المعالج الدقيق عن طريق خطوط التحكم Control Bus

انظمة النواقل (الخطوط) Bus System :

لكي نتعرف على كيفية نقل المعلومات بين المعالج الدقيق و الوحدات الاخرى مثل الذاكرة و وحدة الادخال والاخراج , فيجب ان يكون هناك ناقل (خطوط نقل) تقوم بنقل هذه المعلومات تسمى Bus , وتعرف عملية نقل المعلومات والاشارات بتقنية النقل اي تتصل جميع وحدات الحاسب الدقيق ببعضها بالتوازي (In parallel) وبذلك تنقل الاشارات والمعلومات الى جميع الوحدات في نفس الوقت . يجب ان يكون التحكم في حركة هذه المعلومات و الاشارات عن طريق عنصر واحد فقط يستطيع وضعها في الترتيب الصحيح على خطوط النقل وهذ العنصر هو وحدة المعالج المركزيه (CPU) او ما نسميه بال (Microprocessor) .



شكل يوضح معمارية المعالج 8085

توجد ثلاث انواع من خطوط النقل كما مبين بالشكل اعلاه وهي:

1. خطوط العناوين Address BUS :

لتحديد الموقع الذي يجب ان ترسل اليه المعلومات او نستقبل منه مثلا موقع ذاكرة معين او جهاز ادخال وتكون هذه الخطوط خارجه من المعالج الى الاجهزة الخارجية وليس العكس.
في جميع شرائح ال 8-Bits والتي نحن بصدد الكلام عنها يكون عدد اطراف مسار العنوان Address Bus يساوي 16 مسار لذلك فان مقدار الذاكرة التي يتعامل معها مثل هاذ المعالج يساوي 2^{16}=65536 بايت اي ما يعادل 64KB باعتبار ان كل 1KB=1024Byte .

2. خطوط البيانات Data Bus

تعمل هذه الخطوط في اتجاهين حيث يستطيع المعالج ارسال او استقبال معلومات من والى الوحدات المختلفة و هو خاص بنقل البيانات.فبمجرد ان يحدد المعالج المكان الذي يريد التعامل معه عن طريق العنوان الذي وضعه على Address Bus يقوم المعالج باخراج او استقبال المعلومة نفسها على او من مسار اخر وهو Data Bus .

ان عدد البيانات التي تعرف بها كل شريحة يكون على حسب عدد ال Bits او اطراف خطوط البيانات , فشرائح ال 8-bits سميت كذلك لان لها خطوط بيانات مقداره 8-bits . توضع اي معلومة عادة في صورة الخانات او البتات وكل بت او خانة تاخذ قيمة الواح او الصفر حيث ان الواحد يمثل بجهد معين ويمث الصفر بجهد اخر.كل 8-bits = Byte . عندما يتعامل المعالج مع الذاكرة فان وحدة التعامل بينهما تتوقف على عدد خطوط البيانات لان كل بت تنتقل في خط منفصل.
في حالة الشرائح ال 8-bits فان اي معلومة تنتقل من والى المعالج لابد ان تكون بهذا الحجم , اما اذا كانت اكبر فيتم نقل المعلومة على اكثر من مره بحيث في كل مره يقوم بنقل 8-bits .
في حالة الشرائح 16-bits يتكون وحدة التعامل في نقل المعلومة هي 16-bits وهذا ما يبرر سرعتها مقارنة مع شرائح ال 8-bits وهكذا فكلما زادت عدد بتات خطوط البيانات كلما زادت السرعة في التعامل مع الذاكرة والبتالي سرعة تنفيذ العمليات.

3. خطوط التحكم Control Bus

عن طريق هذه الخطوط يتحكم المعالج بجميع الوحدات .حيث ان عن طريق هذه الخطوط يخبر المعالج اي جهاز من الاجهزه المحيطه الذي تم تحديد عنوانه عن طريق خطوط العناوين عن الغرض من هذا التعامل .

مثلا في حالة تعامل المعالج مع الذاكرة قد يكون الغرض هو للقراءة مثلا , اي استقبال معلومة من الذاكرة, في هذه الحاله فان ال Microprocessor يرسل اشارة على خط التحكم $\overline{MEMR}$, Memory Read تعرف من خلاله الذاكرة ان الغرض من التعامل هو قراءه فترسل المعلومة المطلوبة على خطوط البيانات ليتلقاها المعالج . اما في حالة تعامل المعالج مع الذاكرة لغرض الكتابة , اي ارسال معلومة الى الذاكرة, في هذه الحاله فان ال Microprocessor يرسل اشارة على خط التحكم $\overline{MEMW}$, Memory Writeتعرف من خلاله الذاكرة ان الغرض من التعامل هوالكتابة فتتلقى المعلومة على خطوط البيانات.

هناك ايضا خطان للتحكم بنفس الطريقة للتعامل مع بوابات الادخال والاخراج . هناك ايضا خطوط المقاطعه Interrupt التي بها تتم مقاطعة اي برنامج يجري تنفيذه وخطوط المسك HOLD التي يتم فصل المعالج عن المسارات لاغراض اخرى.

في المعالج الرقمي 8085 فان خطوط التحكم تتكون من اربعة خطوط هي:

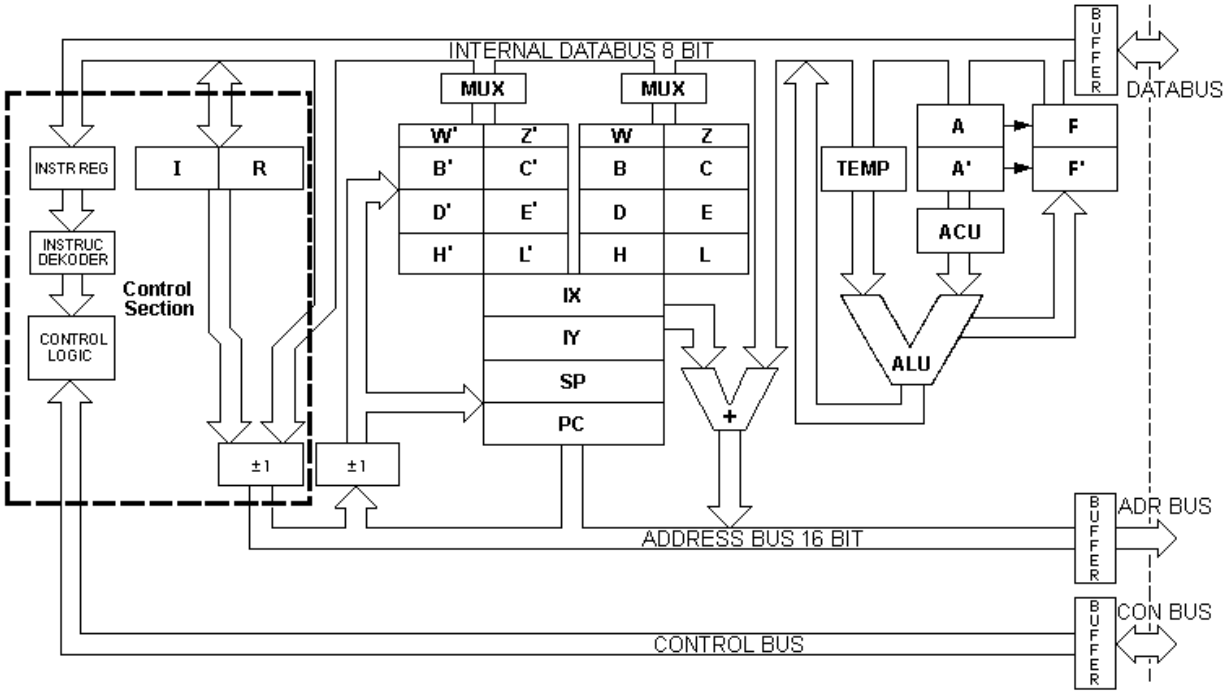
- ❖ قراءة من الذاكرة Memory read .
- ❖ كتابة فب الذاكرة Memory write .
- ❖ قراءة من جهاز ادخال Input device read .
- ❖ كتابة في جهاز اخراج Output device write .

كما يبين الجدول التالي الحصول على خطوط التحكم الاربعة للمعالج 8085 :

	طرف 34 IO/M	طرف 31 WR	طرف 32 RD
IOR	1	1	0
IOW	1	0	1
MEMR	0	1	0
MEMW	0	0	1

المعالج الرقمي Z80

تم تصميم المعالج z-80 من قبل شركة zilog بعد إدخال بعض التطورات على المعالج8080/85. ان تركيب شريحة Z80 هو نفسه تركيب شريحة Intel 8085 من حيث وجود المسجلات الاساسية مثل عداد البرنامج ومسجل ومشفر الاوامر ومسجل التراكم وعدد من المسجلات العامه و وحدة الحساب و المنطق و وحدة التجكم والتزامن , لكن هناك بعض الاختلافات عن هذا الاساس و تتمثل هذه الزيادات في حالة Z80فيما يلي :



نظرة شاملة على اطراف المعالج Z80

1. معظم مسجلات الشريحة Z80 تم مضاعفتها فهناك مثلا مسجلين للتراكم A و A1 ومسجلين للحاله SR و SR1 و جميع المسجلات العامه تم مضاعفتها ايضا مما يعطي المعالج ميزه في عملية البرمجه .
2. تم زيادة المسجلين IX و IY وكل منهما 16-BITS وهذان المسجلان يستخدمان في طرق مختلفة لعنونة الذاكره وتعزيزها بايعازات خاصة .
3. تم زيادة عداد الانعاش الذاكرة والذي يستخدم في عملية انعاش الذاكرة الدينامكيه تحتاج لعملية تجديد او انعاش او اعادة تخزين لمحتوياتها بعد فترات زمنية محددة والا فانها تفقدها بعد زمن مقداره بعض ملي ثانية .
4. المعالج Z80له عدد اطراف مساوي لعدد اطراف المعالج 8085 وهي 40-PIN لها تقريبا نفس الوظائف .
5. مسارات نقل البينات الغير متعددة (معالج 8085 يعين إشارت متعدد في مسار نقل البيانات).

برمجة المعالج 8085

يمكن لاي شخص ان يخبر الحاسب بما سيعمله عن طريق سلسلة من شفرات الاوامر التي تعرف باسم البرنامج ولكن يستطيع المعالج تنفيذ البرنامج الذي يكتب اوامره بشفرة ثنائية فقط تسمى لغة الاله Machine Language , ولما كانت عملية البرمجة باستخدام الشفرة الثنائية عملية شاقة لذا نشات و تطورة لغات البرمجة ,وتوجد برامج جاهزة لتحويل اوامرلغة البرمجة الى الشفرة الثنائية . ومنها لغة التجميع Assembly Language حيث يحدد اسم رمزي فريد لكل امر للحاسب وبعد تحويل هذه الرموز او الاوامر من لغة التجميع الى الالة وهي تمثل (0,1) فقط .

لغات التجميع Assembly Languages هي مجموعة من اللغات ذات المستوى المنخفض (بمعنى أنها مصممة لتتعامل مع الحاسوب أكثر من كونها مصممة ليتم البرمجة بها) تستخدم في برمجة أجهزة الكمبيوتر، المعالجات الدقيقة.Microprocessor وتقوم تلك اللغات بتحويل الكود والثوابت اللازمة لبرمجة بناء معين من وحدات المعالجة المركزيةCPU ، من شكله المعتمد على الرموز Symbolic إلى شكل آخر رقمي يسمى "كود الآلة.Machine Code "هذا التحويل/التمثيل Representation يتم تعريفه عادة عبر الشركات المصنعة للأجهزة، ويعتمد على مجموعة من الاختصارات التي تساعد المبرمجين على تذكر تعليمات البرمجة والسجلات Registers المستخدمة في عمليات البرمجة بسهولة. وهناك لغة تجميع محددة لكل بناء حاسوبي بعكس معظم لغات البرمجة عالية المستوىHigh-Level ، التي عادة ما تعمل مع معظم أنظمة الحاسوب.

وتستخدم أداة برمجية تسمى "المجمِع" Assembler في ترجمة السطور والتعليمات Instructions عبر لغة التجميع إلى "كود الآلة" التي يتم التعامل معها.ويقوم المجمع بتنفيذ ترجمة تماثلية (مثال: سطر مكتوب بلغة التجميع يتحول لسطر مكتوب بلغة الآلة، وهكذا -يسمى One-to-One Mapping) للجمل/السطور المخزنة في ذاكرة الحاسوب ويحولها لتعليمات وبيانات تفهمها الآلة Machine.وهذا يختلف عما يحدث في اللغات عالية المستوى، حيث يتم عادة ترجمة كل جملة إلى عدة أوامر تفهمها الآلة Machine Instructions.

وتقدم العديد من المجمعات المتطورة Assemblers إمكانيات وآليات إضافية تسهل: تطوير البرامج، التحكم في عملية التجميع، والمساعدة في اكتشاف وتصحيح الأخطاء البرمجية Debugging.

ولبرمجة اي معالج لابد من دراسة مجموعة الاوامر الخاصة به ولكي نسهل دراسة هذه الاوامر سنقسمها الى وحدات من حيث الوظيفة التي يؤديها كل امر وهي كالتالي :

1. مجموعة اوامر الانتقال Transfer instruction

يقوم اي امر من اوامر هذه المجموعة بنقل المعلومات من مكان الى اخر . ولكاتبة اي امر يجب ان يكون هناك طرفين اساسين وهما ال Source وهو المكان الذي تاتي منه المعلومة والذي قيد يكون مسجلا او قيد يكون مكان من اماكن الذاكرة . الطرف الثاني هو الDestination وهو الوجهه او الهدف والذي قيد يكون مسجلا او موقع من مواق الذاكره.
من اهم اوامر النقل المستخدمة لبرمجة المعالج 8085 هي:

الايعاز	المختصر
MOV	Move data
MVI	move immediate data
LXI	Load
LDA	Load accumulator direct
STA	Store accumulator direct
LHLD	Load H-L pair direct
SHLD	Store H-L pair direct

- **الامر MOV**

يكون بالشكل التالي

MOV Rd,Rs
MOV M,Rs
MOV Rd,M

ومعنى هذه الامر هو (انقل او حرك) المعلومة الموجودة في المصدرM, Rs الى الهدف Rd,M مع بقاء قيمة المصدر كما هي اي بعبارة اخرى انه يقوم بعملية نسخ .حيث ان المصدر و الهدف من الممكن ان يكون مسجل register او قد يكون موقع ذاكرة Memory Address. من الملاحظ ان قيم مسجل الاعلام Flag register لا تتاثر عند تنفيذ هذا الامر .

الشفرة الثنائية تكتب كالاتي :

01ddd sss

حيث ان : dd هي الشفرة الثنائه للهدف سواء كان مسجلا او مو موقع ذاكره.
و ss هو الشفرة الثنائيه للمصدر الذي قد يكون مسجلا او موقع ذاكرة .

مثال :
افرض ان قيمة المسجل B = 03 وان المسجل A يحمل قيمة 01 فبعد تنفيذ الايعاز التالي تكون قيمه المسجلين A= 03 و الB=03

MOV A,B

اذا اردنا تحويل هذا الامر الى لغة الالة فاننا سوف نقوم يتحويله الى الصيغه الثانيه بحيث نرفع كل رمز و نضع الصيغه الثانية المقابلة له .
وبذلك تكون الشفرة الثنائيه للايعاز السابق هي 01111000
اما الشفرة بال Hexadecimal فهي 78H

مثال:

MOV M,A

هنا سيقوم الامر بنقل محتويات المسجل A وهو المصدر الى موقع الذاكرة M والذي يمثل الهدف .

ملاحظه:

في جميع الاوامر التي تتعامل مع الذاكرة بهذاالشكل يكون العنوان موجودا في زوج المسجلات HL اي ان محتويات المسجل A ستذهب الى بايت الذاكرة التي يوجد عنوانها في المسجلين H,L . تسمى هذه الطريقه بطريقة التعامل غير المباشر مع الذاكرة .

وبذلك تكون الشفرة الثنائيه للايعاز السابق هي 01110111
اما الشفرة بال Hexadecimal فهي 77H

مثال:

MOV B,M

بحيث تكون الشفرة الثنائيه للايعاز هي 01000110
اما الشفرة بال Hexadecimal فهي 46H
حيث يقوم بنقل محتويات الذاكرة الى المسجل B

- **الامر MVI**

وتكون الصيغه العامه له كالاتي

MVI ddd,data8

هذا الامر يضع المعلومة المكونه من 8-bits في الهدف ddd .حيث ان الهدف قد يكون مسجلا او موقع ذاكرة . هذا الامر يتكون دائما من بايتان واحدة هي شفرة الامر operation code واختصارها op code والبايت الاخر هو المعلومه (data8), و بذلك تكون الصيغه العامه
للشفرة الثنائيه كالاتي:

00ddd110
data8

حيث تستبدل الddd بشفرة المسجل او موقع الذاكرة المراد وضع المعلومة فيها

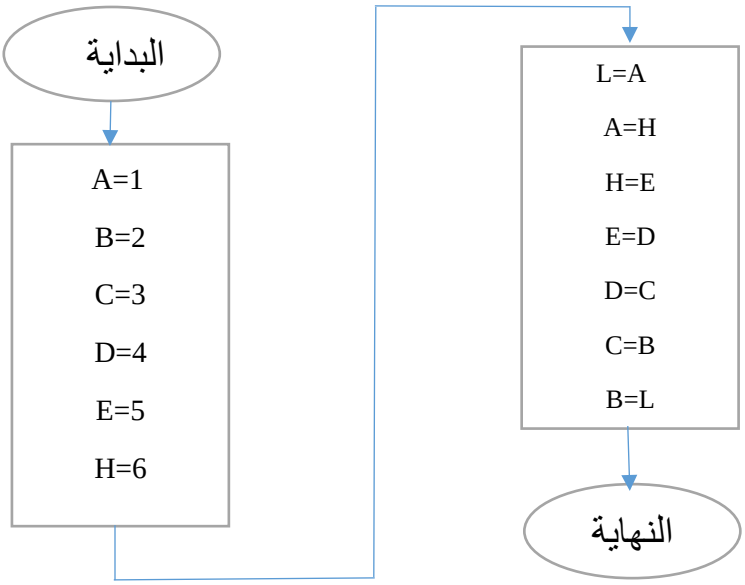
مثال :

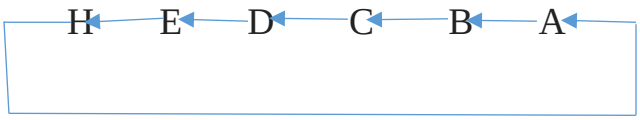
MVI B, 53

الشفرة الثنائيه هي : 00000110
01010011

مثال :

اكتب برنامج يقوم بتحميل المسجلات A,B,C,D,E,H بالمعلومه الفورية 01,02,03,04,05,06 على التوالي ومن ثم تقوم بعمل ازاحة دوارانية لهذه المحتويات .





ازاحة محتويات مجموعة من المسجلات ازراحة دورانية من اليمين الى اليسار

والبرنامج يكون كالتالي :

```
E000    MVI A,01
E001
E002    MVI B,02
E003
E004    MVI C,03
E005
E006    MVI D, 04
E007
E008    MVI E,05
E009
E00A    MVI H,06
E00B
E00C    MOV L,A
E00D    MOV A,H
E00E    MOV H,E
E00F    MOV E,D
E010    MOV D,C
E011    MOV C,B
E012    MOV B,L
```

- **الامر LXI**

و يعني تحميل السجلات المزدوجه الفوري Load Extended Immediate

الصورة العامة لهذا الامر هي

LXI rp,data16

حيث ان المقصود بال rdهو زوج المسجلات .
الشفرة الثنائيه تكون كالتالي:

00rp0001
البايت ذات القيمه الصغرى من المعلومة data16
البايت ذات القيمه العظمى من المعلومة data16

يقوم هذا الامر بتحميل زوج المسجلات الذي توضع شفرته بدلا من rp بالمعلومة الفورية data16 بالمعلومه الفورية ذو ال 16-bits . حيث ان حرف I الموجود في صورة الامر يعني فوري Immediate .كما ذكرنا سابقا فان هناك اربع ازواج من المسجلات للمعالج 8085 وهي SP,HL,DE,BC بحيث ان البايث ذات القيمه العظمى تخزن في H,D,B او النصف الاعلى من المسجل SP اما البايت ذات القيمة الصغرى فتذهب الى المسجل L,E,C او النصف الاول من المسجل SP .

مثال :

LXI H,2C3A

الشفرة الثنائيه لهذا الايعاز هي :

00100001
00111010
00101100

الشفرة الستعشرية هي :

21
3A
2C

حيث 2Cسوف تخزن في المسجل H بينما 3A سوف تخزن في الL .

مثال :

LXI B, 90FFH

عند تنفيذ الامر السابق تكون النتيجه كالتالي :

B=90H
C=FFH
, OR
BC=90FFH

الامران STA و LDA

الامر STA يعني خزن محتويات المركم Store Accumulator اما ال LDA فتعني حمل المركم Load Accumulator والصيغه العامه لكتابة كل امر هي :

STA addr

محتويات المسجل A تذهب الى العنوان addr

LDA addr

محتويات العنوان (addr) تذهب الى المركم .

حيث ان الصورة الستعشريه للامر STA هي :

32

البايت ذو القيمه الصغرى من العنوان Addr

البايت ذو القيمه العظمى من العنوان Addr

و الصورة الستعشريه للامر LDA هي :

3A

البايت ذو القيمه الصغرى من العنوان Addr

البايت ذو القيمه العظمى من العنوان Addr

مثال :

افرض ان A= FFH , ونفذ الايعاز الثالي :

STA E100H

عند تنفيذ الايعاز السابق فان موقع الذاكرة E100H=FFH

مثال :

افرض ان موقع الذاكرة E101=4CH

LDA E101H

عند تنفيذ الايعاز السابق فان المسجل المركم سوف يحمل بقيمة موقع الذاكرة حيث A=4CH

مثال :

اكتب برنامج يقوم بخزن 8-bits data الى موقع الذاكره بحيث تكون القيمة المخزونه 52H وموقع الذاكره هو 4000H .

```
1. Program 1:
2. MVI A, 52H : "Store 32H in the accumulator"
3. STA 4000H  : "Copy accumulator contents at address 4000H"
4. HLT        : "Terminate program execution"

1. Program 2:
2. LXI H  : "Load HL with 4000H"
3. MVI M,52H : "Store 52H in memory location pointed by HL register
   pair (4000H)"
4. HLT    : "Terminate program execution"
```

اكتب برنامج يقوم بتبديل محتويات عنوان الذاكرة 2000 و 4000 ببعضهما .

```
1. Program 1:
2. LDA 2000H   : "Get the contents of memory location 2000H into
   accumulator"
3. MOV B, A    : "Save the contents into B register"
4. LDA 4000H   : "Get the contents of memory location 4000Hinto
   accumulator"
5. STA 2000H   : "Store the contents of accumulator at address
   2000H"
6. MOV A, B    : "Get the saved contents back into A register"
7. STA 4000H   : "Store the contents of accumulator at address
   4000H"
```

الامران LHLD و SHLD

والصورة العامة للايعاز LHLD هي :

LHLD addr

و يعني محتويات (addr+1)تذهب الى H و محتويات (addr)تذهب الى L

حيث ان الصورة الستعشريه للامر LHLD هي :

2A

البايت ذو القيمه الصغرى من العنوان Addr

البايت ذو القيمه العظمى من العنوان Addr

مثال : اذا كانت مواقع الذاكرة تحمل القيم التاليه :

M(1234H)=00H , M(1235H) = 50H

فان تنفيذ الايعاز LHLD 1234H يعطي

HL=5000

والصورة العامة للايعاز SHLD هي :

SHLD addr

يكون هذا الايعاز مشابها للايعاز LHLD عدا ان البايتين في السجل المزدوج HL تخزن في العنوان المحدد والعنوان الاعلى التالي .

و الصورة الستعشريه للامر SHLD هي :

22

البايت ذو القيمه الصغرى من العنوان Addr

البايت ذو القيمه العظمى من العنوان Addr

مثال:

افرض ان قيمة ال H= 89 و قيمة L=76 و موقعي الذاكره E100=FF و E101=EE . فما هي النتيجة بعد تنفيذ الايعاز التالي :

SHLD E100

بعد تنفيذ الامر السايق فان النتيجة تكون ان الموقعي الذاكرة سوف تاخذ قيم المسجلات بحيث E100=76 و E101=89

اما عند نتفيذ الايعاز :

LHLD E100

فانالنتيجة تكون H=EE وL=FF

الامر XCHG:

صيغته العامه : XCHG

يقوم هذا الامر بتبديل محتويات المسجلات المزدوجه DE, HL ببعضها .

مثال:

اذا كان

DE=1234H

HL=5678H

فعند تنفيذ :

XCHG

تكون النتيجة

DE=5678H

HL=1234H

2. مجموعة اوامر الحساب Arithmetic Instructions

حيث يكون مسجل المراكم A طرفا في عملية حسابية او منطقية وتسجل نتيجة اي عملية في المركم A . تتميز هذه المجموعة من الاوامر عن غيرها من الايعازات او الاوامر بانها عندما يتم تنفيذ امر من هذه الاوامر فان الاعلام الموجودة في مسجل الحالة SR تتاثر بتيجة هذه العملية .

الايعاز	المختصر
ADD	Add
SUB	Subtract
ADI	Add the immediate to the accumulator
SUI	Subtract the immediate from the accumulator
ADC	Add register to the accumulator with carry
SBB	Subtract the source and borrow from the accumulator
INR	Increment the register or the memory by 1
INX	Increment register pair by 1
DCR	Decrement the register or the memory by 1
DCX	Decrement the register pair by 1

الامر ADD

الصيغة العامه للامر

ADD R
ADD M

حيث يقوم هتذ الايعاز بعملية جمع محتويات المسجل R او محتويات موقع الذاكرة مع محتويات المسجل المركم A , ويخزن حاصل الجمع في المسجل المركم . تنفيذ هذا الايعاز يوثر على مسجل الحالة .

مثال :

A= 11110001 , E = 00001000

نفذ الامر

ADD E

النتيجه تكون A= 11111001

الامر ADC

الصيغة العامه للامر

ADC R
ADC M

حيث يقوم هتذ الايعاز بعملية جمع محتويات المسجل R او محتويات موقع الذاكرة مع محتويات المسجل المركم A , زائدا علم الحمل في مسجل الحالة , ويخزن حاصل الجمع في المسجل المركم .

مثال :

A= 10000011 , E = 00010010 , CY=1

نفذ الامر

ADC E

عند تنفيذ الامر السابق يكون الحل كالتالي :

```
 10000011
 00010010
      +1
 ========
 10010110
```

وتكون محتويات المسجل المركم و الحمل كالتالي :

CY=0 A= 10010110

مثال :

اكتب برنامجا يجمع العددين 700 و 900 ويخزن الجواب في المسجلات H و L .

الحل :

ان العددين 700 و 900 يمكن تمثيلها ببايتين كما مبين

700 = 02BCH=0000 0010 1011 1100
900 = 0384H = 0000 0011 1000 0100

البرنامج يكون كالتالي

```
MVI A,00H
MVI B,02H
MVI C, BCH
MVI D,03H
```

```
MVI E,84H
ADD C
ADD E
MOV L,A
MVI A, 00H
ADC B
ADD D
MOV H,A
HLT
```

يكون الجواب كاملا هو 0640H اي ما يقابل 1600 عشري مخونا في المسجلات كالتالي :

H=06H

L=40H

الامر SUB

الصيغة العامه للامر

```
SUB R
SUB M
```

حيث يقوم هتذ الايعاز بعملية طرح محتويات المسجل R او محتويات موقع الذاكرة من محتويات المسجل المركم A , ويخزن حاصل الطرح في المسجل المركم .

مثال :

A= 00001111 , C = 00000001 ,

نفذ الامر

```
SUB C
```

عند تنفيذ الامر السابق يكون الحل كالتالي :

```
  00001111
- 00000001
==========
  00001110
```

وتكون محتويات المسجل المركم و الحمل كالتالي :

CY=0 A= 00001110

مثال : اذا كان محتويات المسجلات كالاتي :

A= 00001100 , C = 00010010 ,

نفذ الامر

SUB C

عند تنفيذ الامر السابق يكون الحل كالتالي :

```
  00001100
- 00010010
==========
 1 11111010
```

لاحظ الاعارة الاخيرة , تحدث هذه الاعارة لان محتويات المسجل C (18 حسب النظام العشري) اكبر من محتويات المسجل المركم A الذي يساوي 12 بالنظام العشري و بذلك تكون محتويات المسجل المركم و الحمل كالتالي :

CY=1 A= 11111010

الامر SBB

الصيغة العامه للامر

SBB R
SBB M

و يمثل عملية طرح مع الاعارة Subtract with Barrowحيث يذهب هذا الايعاز خطوة واحدة ابعد من ايعاز SUB حيث يطرح محتويا المسجل الموصوف او موقع الذاكره و علم ال CY من المسجل المركم

مثال : اذا كان

A= 11111111 , C = 00000010 , CY=1

نفذ الامر

SBB C

يبتدئ هذا الايعاز بجمع المسجل C مع الحمل للحصول على 00000011 ومن ثم طرحها من المسجل المركم .كالتالي

```
  11111111
- 00000011
==========
  11111100
```

وتكون محتويات المسجل المركم و الحمل كالتالي :

CY=0 A= 11111100

الامر ADI

الصيغة العامه للامر

ADI data-8

ويمثل عملية جمع بين محتويات المسجل المركم و المعلومه ذو ال 8 بت

الامر SUI

الصيغة العامه للامر

SUI data-8

و يمثل عملية طرح بين المسجل المركم و المعلومة ذو ال8 بت

مثال :

اكتب برنامجا يطرح 700 من 900 ويخزن الجواب في المسجلات H و L .

الحل :

ان العددين 700 و 900 يمكن تمثيلها ببايتين كما مبين

700 = 02BCH=0000 0010 1011 1100

900 = 0384H = 0000 0011 1000 0100

البرنامج يكون كالتالي

```
MVI A,84H
SUI BDH
MOV L,A
MVI A,03
MVI D,02H
SBB D
MOV H,A
HLT
```

يكون الجواب كاملا هو 00C8H مخزونا في المسجلات كالتالي :

H=00H

L=C8H

الامر INR

INR و يعني الزيادة الذي يقوم بعملية زيادة محتويات السجل او محتويات موقع الذاكرة بمقدار 1 .الصيغة العامه للامر هي :

INR R
INR M

مثال : اذا كانت محتويات المسجل C=8A فعند تنفيذ الامر التالي _

INR C

تكون النتيجة C=8B

الامر DCR

DCR و يعني النقصان الذي يقوم بعملية انقاص محتويات السجل او محتويات موقع الذاكرة بمقدار 1 .الصيغة العامه للامر هي :

DCR R
DCR M

مثال : اذا كانت محتويات المسجل C=8A فعند تنفيذ الامر التالي _

DCR C

تكون النتيجة C=89

مثال :
اكتب برنامج لجمع العددين 23, 25 وخزن الناتج في الموقع 5600H بعد ذلك قم بعملية زيادة الناتج بمقدار 1 و اخزن النتيجة النهائية في المسجل C . ابدء كتابة البرنامج عند الموقع 2000H

2000H	3E	MVI A,17
2001H	17	
2002H	06	MVI B, 2D
2003H	2D	
2004H	80	ADD B
2005H	32	STA 5600
2006H	00	
2007H	56	

2008H	3C	INR A
2009H	4F	MOV C, A
200AH	76	HLT

لاحظ ان ايعازات ADD ,INR, MOV,HLT هي ايعازات ذات بايت واحد اما ايعازات ال MVI فانها ذات البايتين و STA تكون ذات ال 3 بايتات .

مثال:

اكتب برنامج يقوم بعملية جمع رقمين مخزونين في موقعي الذاكرة 4000 و 4001 وخزن النتيجة في موقع الذاكرة 4002 .

```
1. Sample problem
2. (4000H) = 14H
3. (4001H) = 89H
4. Result     = 14H + 89H = 9DH
5.
6. Source program
7. LXI H 4000H : "HL points 4000H"
8. MOV A, M    : "Get first operand"
9. INX H       : "HL points 4001H"
10. ADD M       : "Add second operand"
11. INX H       : "HL points 4002H"
12. MOV M, A    : "Store result at 4002H"
13. HLT         : "Terminate program execution"
```

مثال :

اكتب برنامج يقوم بعملية طرح محتويات موقع الذاكرة 4001 من محتويات موقع الذاكره 2000 ويخزن النتيجه في الموقع 4002 .

```
1. Program –: Subtract two 8-bit numbers
2. Sample problem:
3. (4000H) = 51H
4. (4001H) = 19H
5. Result    = 51H – 19H = 38H
6.
7. Source program:
8. LXI H, 4000H : “HL points 4000H”
```

```
9. MOV A, M     : “Get first operand”
10. INX H        : “HL points 4001H”
11. SUB M        : “Subtract second operand”
12. INX H        : “HL points 4002H”
13. MOV M, A     : “Store result at 4002H”
14. HLT          : “Terminate program execution”
```

2. مجموعة اوامر المنطق Logical Instructions

الامر	المختصر
ANA	AND with A
ORA	OR with A
XRA	XOR with A
ANI	AND with Immediate
ORI	OR with Immediate
XRI	XOR with Immediate
CMA	Complement A
CMP	Compare
CPI	Compare immediate with the accumulator

ANA reg : اجراء عملية AND على محتويات المسجل reg مع محتويات المركم A وتسجيل النتيجه في الركم .
ORA reg : اجراء عملية OR على محتويات المسجل reg مع محتويات المركم A وتسجيل النتيجه في الركم .
XRA reg : اجراء عملية XOR على محتويات المسجل reg مع محتويات المركم A وتسجيل النتيجه في الركم .
ANI data : اجراء عملية AND على قيمة فورية او ثابتة مع محتويات المركم A وتسجيل النتيجه في الركم .
ORI data : اجراء عملية OR على قيمة فورية او ثابتة مع محتويات المركم A وتسجيل النتيجه في الركم .

XRI data : اجراء عملية XOR على قيمة فورية او ثابتة مع محتويات المركم A وتسجيل النتيجه في الركم .
CMA: ايجاد المتمم لمحتويات المركم A.

- بالنسبة المجوعة اوامر المنطق فانها تشمل الايعازات ادناه

ANA (R or M)
ORA (R or M)
XRA (R or M)

حيث تقوم هذه الايعازات بعملية AND , OR, and Exclusive OR بين المسجل المركم وبين المسجل المستخدم او موقع الذاكرة .

اما الايعازات

ANI data8
ORI data8
XRI data8

فانها تقوم بالعملية السابقه ولكن هذه المرة بين محتويات المسجل المركم و بين المعلومه ذو ال 8 بت

- بالنسبة للايعاز CMA فانه يعطي المتتم للمسجل المركم .
- لدينا ايضا ايعازات المقارنة وهي CMPو CPI حيث يقوم الايعاز الاول بعملية مقارنة بين محتويات المسجل المستخدم او مواقع الذاكرة مع محتويات المسجل المركم وتكون صيغته كالتالي:

CMP R
CMP M

اما ايعاز ال CPI فيقوم بعملية مقارنة بين محتويات المسجل المركم وبين المعلومه الفورية ذو ال8 بت وتكون صيغة الامر كالتالي:

CPI data-8

مثال

اكتب برنامج يقوم بحساب الممتم لرقم مخزون في موقع الذاكرة 4400 و يخزن النتيجة في الموقع 4300 .

```
1. Sample problem:
2. (4400H) = 55H
3. Result = (4300H) = AAH
4. Source program:
5. LDA 4400H : "Get the number"
6. CMA       : "Complement number"
7. STA 4300H : "Store the result"
8. HLT       : "Terminate program execution"
```

3. مجموعة اوامر القفز Jump Instructions

ان القاعده العامه ان يقوم المعالج بتنفيذ البرنامج حسب ترتيب الاوامر الموجودة فيه من بدايته وحتى النهاية , و لكن هناك بعض الحالات التي تتطلب الخروج عن هذه القاعده كان يطلب منك تنفيذ عملية معينه عدد معين من المرات او عدد لانهائي من المرات .

وهذه الاوامر هي :

الامر	المختصر
JMP	Jump
JZ	Jump if zero
JNZ	Jump if not zero
JM	Jump if negative
JP	Jump if positive
JC	Jump if carry
JNC	Jump if no carry
JPO	Jump if odd parity
JPE	Jump if even parity

يوجد نوعان من اوامر القفز :

a) **القفز غير المشروط Unconditional jump**

في هذا النوع ينتقل المعالج بعملية التنفيذ الى المكان الجديد دون شرط او قيد . و هناك امر واحد فقط من اوامر القفز للمعالج 8085 يقوم بهذه العملية وهو ابعاز او امر JMP حيث يكتب كالاتي :

JMP addr

حيث ان عند تنفيذ هذا الامر يوضع العنوان addr الذي سيتم القفز اليه في عداد البرنامج فيصبح الامر الموجود عند هذا العنوان هو الامر الذي عليه الدور في التنفيذ .

ان القفز باستخدام هذا الامر قد يكون الى الامام في البرنامج او قد يكون الى الخلف . اذا كان القفز الى الامام سينتج عن ذلك وجود جزء من البرنامج لن ينفذ على الاطلاق وهو الجز الذي يبن الامر JMP ويبن الامر الذي سيتم القفز اليه . اما في حالة القفز الى الخلف فسينتج نتيجة ذلك ما يسمى بالحلقة اللانهائية والتي سوف يستمر المعالج بتنفيذها الى المالانهاية.

b) **القفز المشروط Conditional jump**

حيث ينقل المعالج بعملية التنفيذ الى مكان جديد بشرط ان يتحقق هذا الشرط وبخالافه لاتتم عملية الانتقال ويستمر تنفيذ البرنامج وحسب تسلسله الطبيعي حيث سينفذ الامر الذي بعد امر القفز مباشرة .

ان شروط القفز توضع دائما على الاعلام في مسجل الحاله (الاعلام) , من هذه الاوامر :

JZ addr : اقفز اذا كانت النتيجة صفرا
JNZ addr : اقفز اذا كانت النتيجة ليست صفرا.
JM addr : اقفز اذ كانت النتيجة سالبة .
JP addr : اقفز اذ كانت النتيجة موجبة .
JC addr : اقفز اذا كان هناك حمل .
JNC addr : اقفز اذ لم يكن هناك حمل .
JPO addr : اقفز اذا كان البارتي فردي .
JPE addr : اقفز اذا كان البارتي زوجي.

مثال :

```
        MVI A,FEH
REPEAT : ADI 01H
        JNC REPEAT
        MVI A, C4H
        JC ESCAPE
          .
          .
ESCAPE:  MOV L,A
```

مثال :

يبن ما هو عمل البرنامج التالي :

```
        MOV E,00H
LOOP: INR E
        MOV A,E
        CPI FFH
        JNZ LOOP
```

- يستخدم المسجل E كعداد اذ انه يبدا عند ال 0 . تاخذ قيمه السجل يازياده بمقدار 1 و تنقل القيمه الى المسجل المركم ويتم بعد ذلك مقارنته بالقيمه 255 , هنا يكون علم الصفر Z=0 لان قيمه ال FFH, 01H غير متساويين فيضطر البرنامج للعودة الى موقع الذاكرة ذو العنوان LOOP .

- يستمر المرور خلال الدارة حتى ينفذ INR و MOV العدد 255 مرة للحصول على A=FFH
- في هذه الحالة وعند يكون علم الصفر يساوي واحد وبذلك يتوقف البرنامج
- يجب ملاحظه ان البرنامج سوف يستمر بالمرور في هذه الدورة 255 مرة قبل الهبوط خلال JNZ لذلك نستطيع استخدام هذا البرنامج لتكوين تاخير زمني .

4. اوامر الادخال و الاخراج Input/ Output Instruction

a. امر ادخال البيانات IN

وهذه الامر يعني ادخال البيانات الموجودة عند بوابة الادخال الى المركم وصيغته :

IN Byte

b. امر اخراج البيانات OUT

وهذا الامر يعني اخراج البيانات الموجودة في المركم الى بوابة الخروج و صيغته :

OUT Byte

الاسمبلي	الشفرة
MOV ddd,sss	01dddsss
MVI ddd,data-8	00ddd110 Data8
LXI rp,data16	00rp0001 Data16
STA addr.	32 Addr 16
LDA addr.	3A Addr.16
XCHG	EB
ADD sss	10000sss
ADC sss	10001sss
ADI data8	11000110 Data8
SUB sss	10010sss
SBB sss	10010sss
SBI data8	DE Data8
SUI data8	D6 Data8
INR ddd	00ddd100
INX rp	00rp0011
DCR ddd	00ddd101
DCX rp	00rp1011
ORA sss	10110sss
ORI data8	F6 Data8
ANA sss	10100sss
ANI data8	11100110 Data8
CMA	2F
CPI data8	FE Data8
JMP addr	C3 Addr.16
JNC addr	D2 Addr.16
JC addr	DA Addr.16
JM addr	FA Addr.16
JZ addr	CA Addr.16
JNZ addr	C2 Addr.16
JP addr	F2 Addr.16
JPE addr	EA

	Addr.16
JPO addr	E2 Addr.16
IN no.	DB Port no.
OUT no.	D3 Port no.
HLT	76