جامعة حماه
الكلية التطبيقية
قسم تقنيات الحاسوب

مقرر تنظيم الحاسوب والبرمجة بلغة التجميع
السنة الأولى

Basic Computer Organization and Design

الفصل الثاني

الدكتور المهندس
أحمد كـردي

1- شفرات التعليمية Instruction Codes :

يعريف تنظيم حاسب: هو جملة المسجلات الداخلية، والتوقيت أو التزامن، وبناء وحدة التحكم ، ومجموعة التعليمات التي يستخدمها.

يتم تنظيم نظام رقمي عن طريق التعليمات الميكروية التي ينفذها على المعطيات المخزنة في المسجلات حيث يمكن للمستخدم إجراء عملية ما بواسطة برنامج Program والذي هو عبارة عن سلسلة من التعليمات التي تنفذ عمليات معينة على معاملات (أو معطيات) بغية الحصول على نتائج نهائية.

تمثل التعليمة بشفرة ثنائية، وتوضع في ذاكرة الحاسب مع سلسلة التعليمات الأخرى المكونة للبرنامج وذلك في منطقة خاصة منعزلة عن منطقة المعطيات

تقسم شفرة التعليمة إلى قسمين:

1- يحدد عمل هذه التعليمة Op Code

2- يحدد المعامل الذي ستجري عليه التعليمة Operand

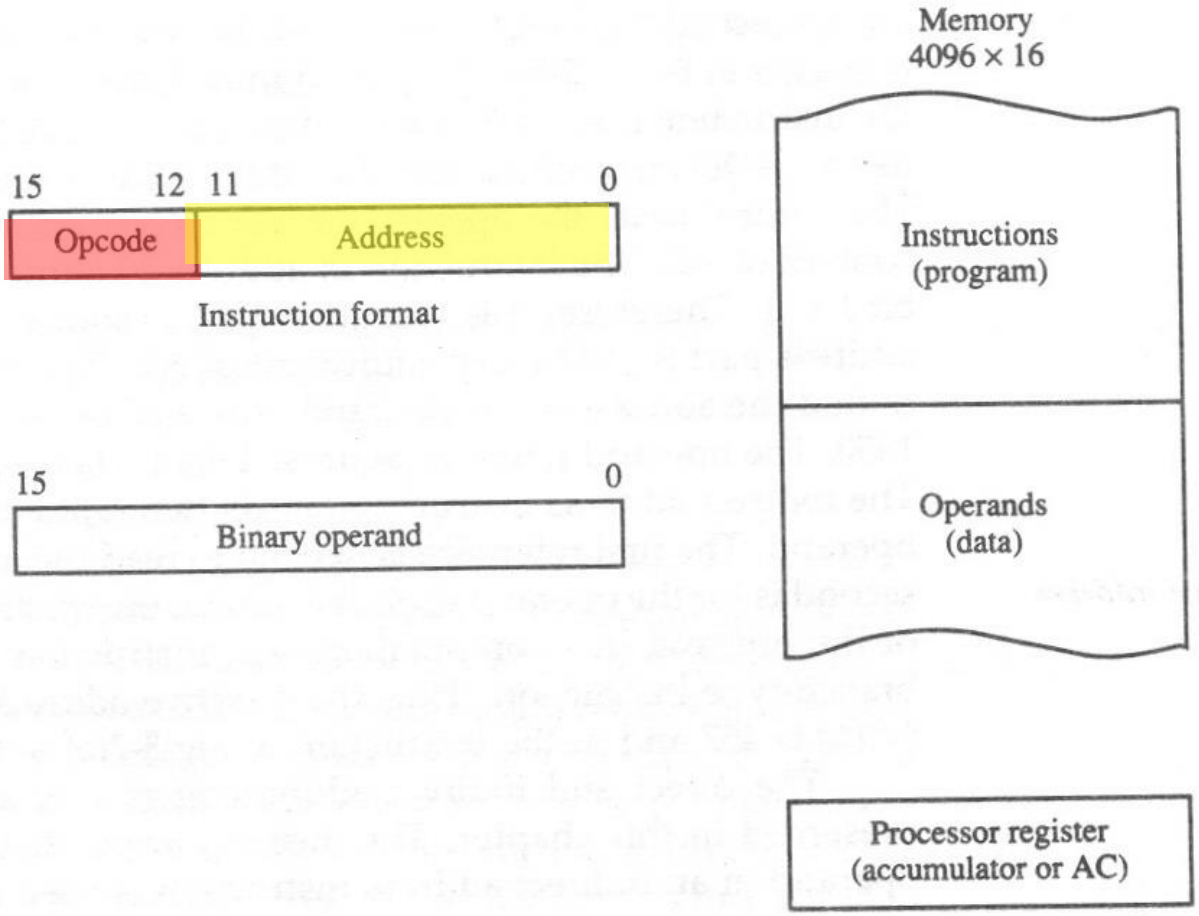
فإذا كان القسم الذي يحدد عمل التعليمة يتألف من n Bit عندها يكون عدد التعليمات في الحاسب هو 2^n أو أقل وإذا كانت التعليمة المنفذة لها معاملات (كتعليمة الجمع ADD)

عندها نفترض أن التعليمة تنفذ على المعامل المشار إليه في شفرة التعليمة كطرف وعلى مسجل المعالج Processor Register يدعى Accumulator بالمراكم ويرمز له اختصارا **AC**

يبين الشكل (1- 4) هذا الأسلوب في التنظيم، حيث تخزن التعليمات في ذاكرة بفترض 4K Word (4096 x 16) أي بخطوط عنونة عددها 12 ،

كذلك يتم تخزين المعاملات والمعطيات حيث ان طول التعليمة هو 16Bit يذهب منها 12 Bit لتمثيل المعامل (طول عنوان الذاكرة) ويبقى 4 Bit لتمثيل شفرة العمل Op Code وبما أن طول المعطيات هو 16 Bit فإن المراكم AC سيكون بنفس الطول .

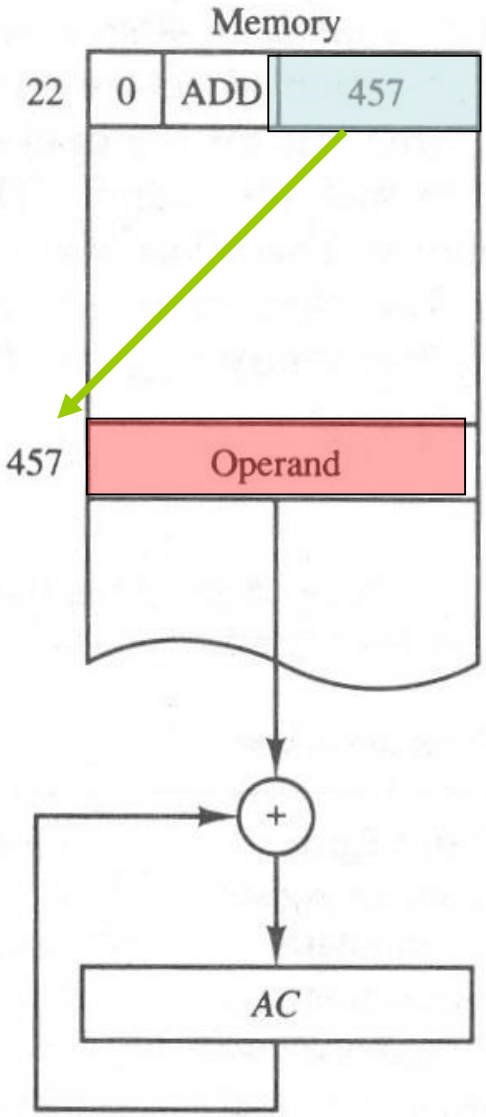
Figure 4-1 Stored program organization.



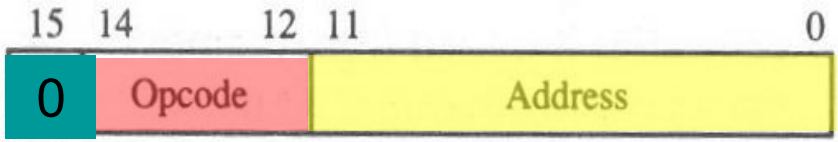
يمكن أن تتم العنونة في الذاكرة عن طريقين :

1- العنونة المباشرة Direct Addressing :

يذكر العنوان بشكل صريح (الشكل 3-2b)



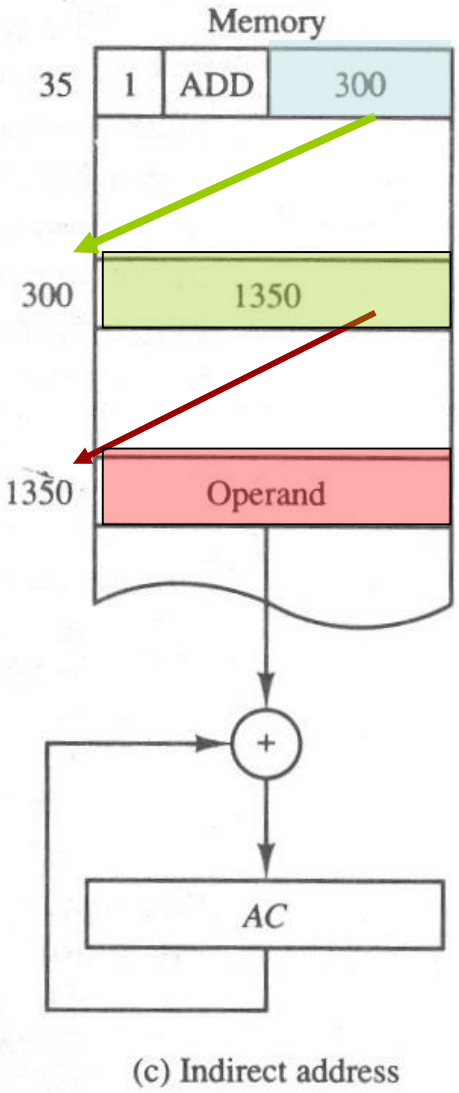
(b) Direct address



(a) Instruction format

Figure 4-1

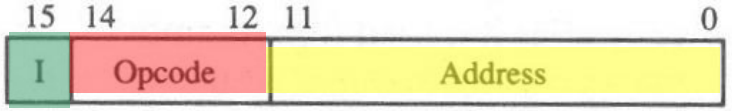
2-العنونة غير المباشرة indirect Addressing : يتم التصريح عن عنوان يحتوي على عنوان آخر، مضمونه يمثل المعطيات المراد تنفيذ التعليمة عليها (الشكل 4-2c)



(c) Indirect address

يمكن في حاسبنا الافتراضي ان نشير للخانة 15 على طريقة العنونة I :

- فإذا كانت هذه الخانة I فالعنونة غير مباشرة
- إذا كانت 0 فالعنونة مباشرة (الشكل 4-2a)



(a) Instruction format

2- مسجلات الحاسب Computer Registers :

تخزن التعليمات في الذاكرة بالتسلسل، وتقوم وحدة التحكم بقراءة التعليمة الموجودة في عنوان محدد ثم تنفيذها،

في الخطوة التالية: تقرأ التعليمة التالية وتنفيذها.

يحتاج هذا النوع من التسلسل إلى عداد يقوم بحساب عنوان التعليمة التالية بعد تنفيذها، ونمثل هذا العداد بمسجل يدعى **عداد البرنامج PC program Counter** وهو يشير إلى عنوان التعليمة المنفذة أي بطول 12Bit.

أيضاً تحتاج وحدة التحكم إلى مسجل يضع فيه شفرة التعليمة لتحليلها وتنفيذها يدعى هذا المسجل **بمسجل التعليمة IR Instruction Reg.** وكون شفرة التعليمة بطول 16Bir لذلك يأخذ نفس الطول .

نحتاج إلى مسجل نضع فيه المعطيات التي يراد معالجتها بعد نقلها من الذاكرة، يدعى **بمسجل المعطيات DR Data Reg.** وهو بطول 16Bit (نفس طول المعطيات)

نحتاج إلى مسجل نضع فيه العنوان الدال على المعطيات في الذاكرة يدعى **بمسجل العنوان Address AR Reg.** وطوله 12 Bit (يعنون الذاكرة المفترضة 4Kword)

نحتاج إلى مسجل نخزن فيه معطيات مؤقتة وهو **مسجل المؤقت Temporary** وهو بطول 16Bit

أخيرا نحتاج إلى مسجلين أحدهما لإدخال حرف يدعى **مسجل الإدخال INPR input Reg.**

والأخر لإخراج حرف هو **مسجل الإخراج OUTR Output Reg.**

كلاهما بطول 8 Bit وهي الشفرة الممثلة لحرف Character

يبين الجدول (1-4) لائحة بهذه المسجلات

Table 4-1 List of Registers for the Basic Computer

Register symbol	Number of bits	Register name	Function
DR	16	Data register	Holds memory operand
AR	12	Address register	Holds address for memory
AC	16	Accumulator	Processor register
IR	16	Instruction register	Holds instruction code
PC	12	Program counter	Holds address of instruction
TR	16	Temporary register	Holds temporary data
INPR	8	Input register	Holds input character
OUTR	8	Output register	Holds output character

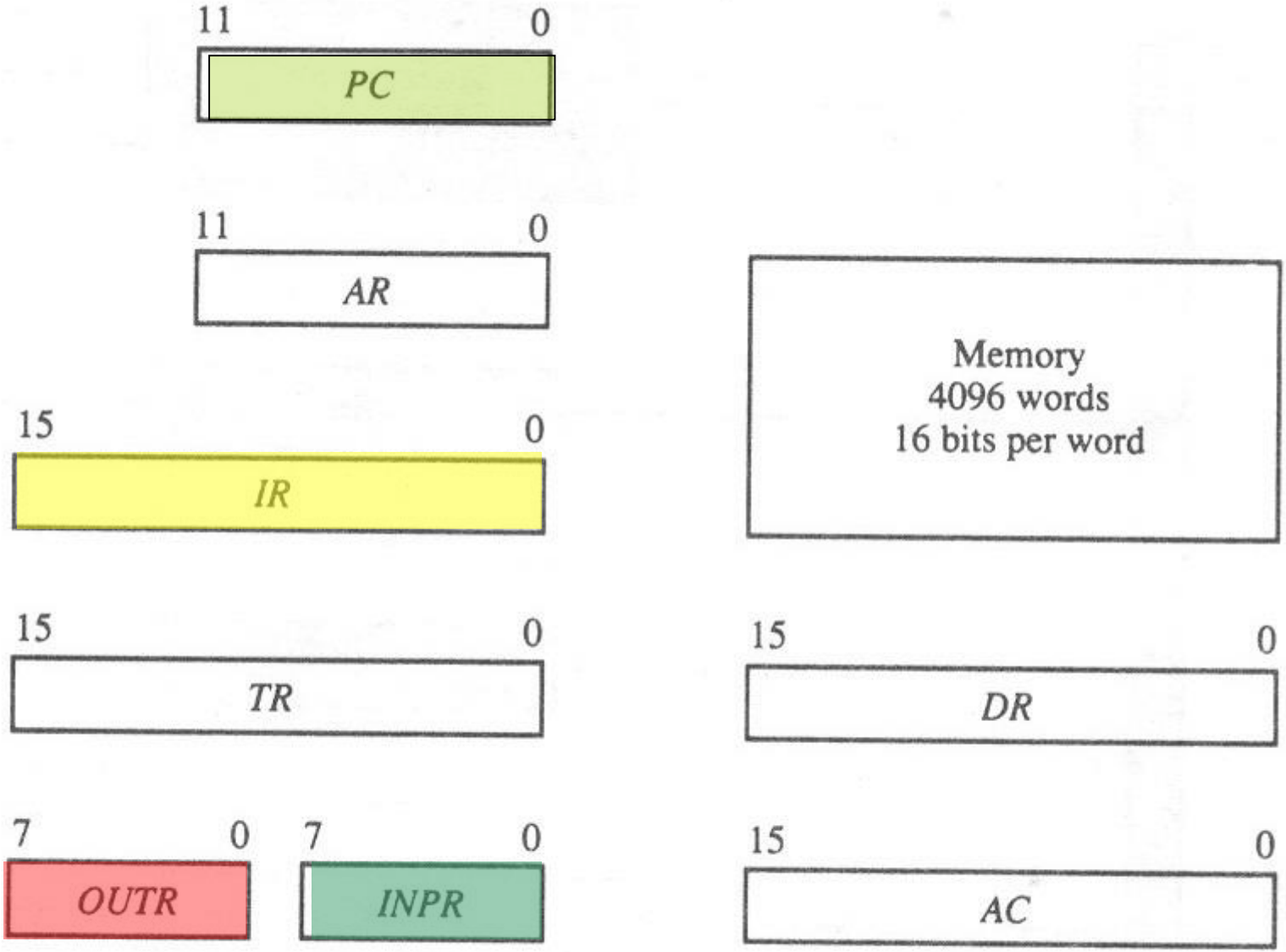


Figure 4-3 Basic computer registers and memory.

نظام الممر المشترك:

نبتعد عند ربط المسجلات والذاكرة مع بعضها البعض عن التوصيل المباشر لأنه يستهلك عددا كبيرا من الخطوط ونستخدم كبديل **الممر المشترك Common Bus** ونضمنه العوازل على كل مسجل وذاكرة بحيث يصبح الربط كالشكل (4-4)

يمكن اختيار المسجلات والذاكرة التي ستضع محتوياتها على الممر باستخدام مشفر اختيار له ثلاثة خطوط تحكم S_0, S_1, S_2

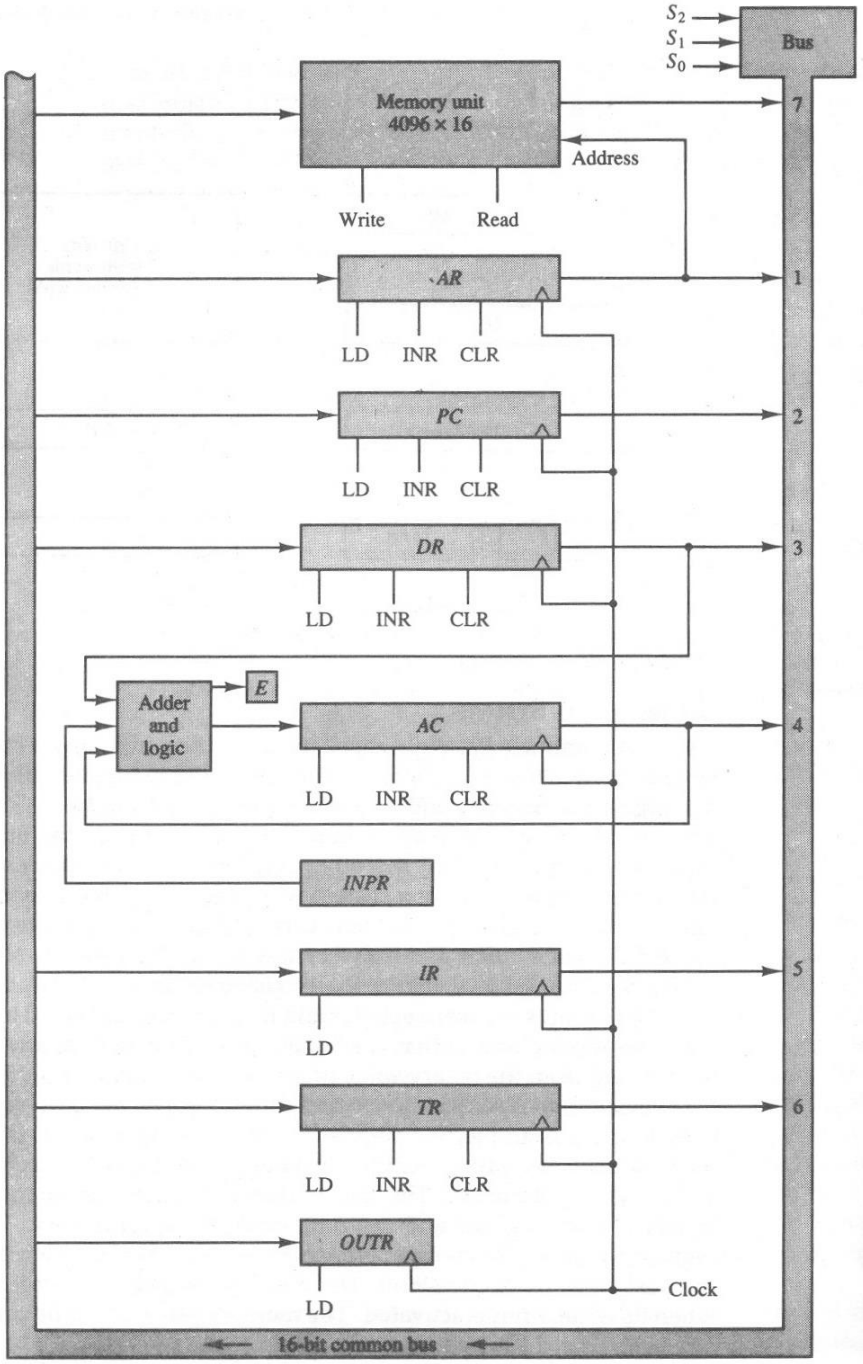


Figure 4-4 Basic computer registers connected to a common bus.

- نوزع مخارج المشفر بحيث حسب قيمة S0,S1,S2 يتم اختيار المسجل أي:

If $S_2\ S_1\ S_0$	= 1 then	select	**AR**
	= 2		**PC**
	= 3		**DR**
	= 4		**AC**
	= 5		**IR**
	= 6		**TR**
	= 7		**Mem**

- كما يتم وضع خط تحكم LD (تحميل Load) على المسجلات باستثناء INPR والذي يمكن المسجل من استقبال المعطيات الموجودة على الممر و وضعها بداخله أما الذاكرة فيكون لها خط تحكم Write لأداء نفس المهمة

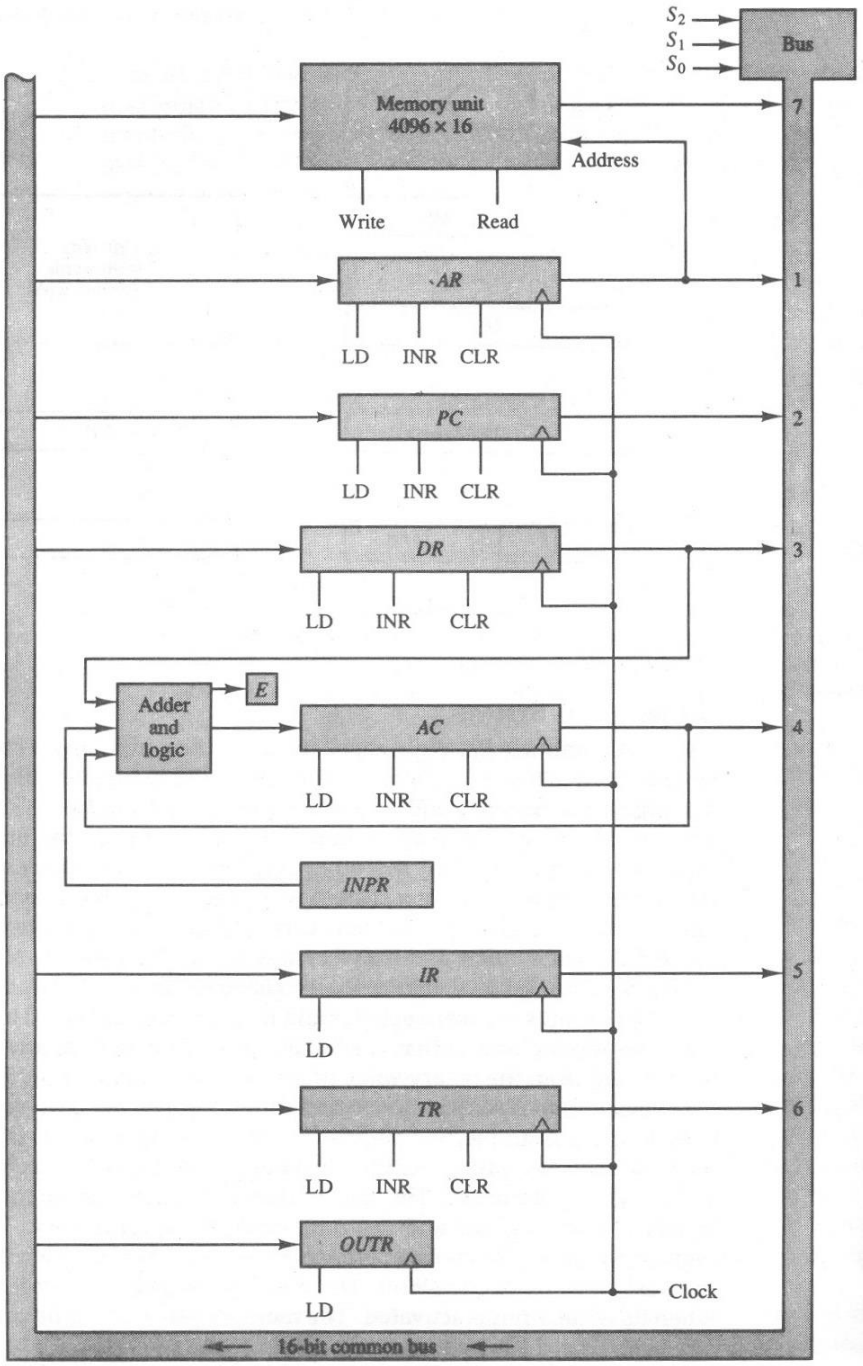


Figure 4-4 Basic computer registers connected to a common bus.

- كما يتم وضع خطين للتحكم INR و CLR لبعض المسجلات تقوم بعمل التصفير CLR وزيادة واحد INR
- يوضع خط Read للذاكرة ليمكنها من وضع محتوياتها على الممر المشترك .

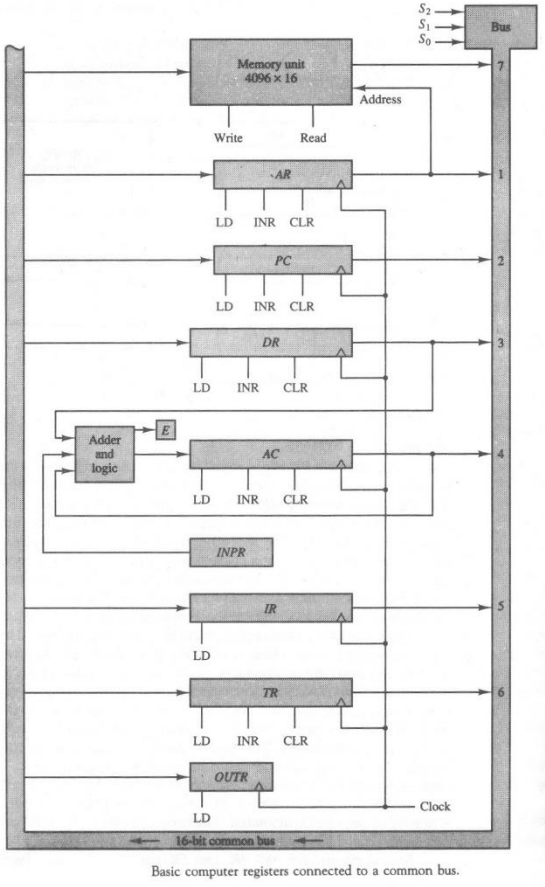
- وتجدر الإشارة ان محتويات AR و PC بطول 12 Bit وبالتالي فإن الممر يتصل بها بـ 12 خط (الأقل أهمية)

- وفي حال إجراء نقل محتويات أحد هذين المسجلين إلى الممر فإن الخانات الأربعة الأكثر أهمية تاخذ حالة 0

- أيضا توصل العناوين للذاكرة مع خرج المسجل AR مباشرة بحيث تتم عنونة الذاكرة عبر هذا المسجل حصرا.

مثال

نورد فيما يلي جدولا يبين مجموعة حالات خطوط تحكم المشفر S2,S1,S1 وعلى أي مسجل سوف يتم تطبيق خط LD ووضعة الذاكرة (قراءة أم كتابة) ووضعية دارة Adder & Logic وما ستؤدي هذه المجموع من عمليات



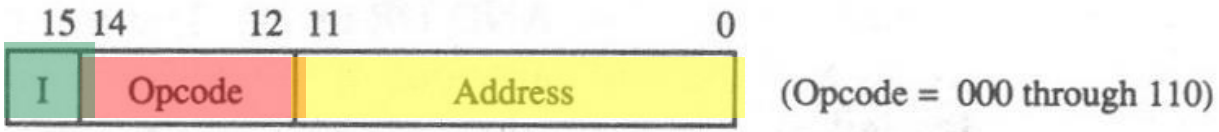
Basic computer registers connected to a common bus.

S	S2	S1	S0	LD of Register	Memory	Adder	Operation
a	1	1	1	IR	Read	-	IR ←M[AR]
b	1	1	0	PC	—	-	PC←TR
c	1	0	0	DR	Write	-	M[AR] ←AC.DR←AC
d	0	0	0	AC	—	Add	AC←AC+DR
e	0	0	0	DR.AC	—	-	DR←AC.AC←DR

3- تعليمات الحاسب Computer Instruction:

بفرض أن للحاسب ثلاثة أنواع من شفرة التعليمة بطول 16Bit :

1- تعليمة تعامل مع الذاكرة Memory Reference Instruction:



(a) Memory – reference instruction

تستخدم 12 Bit للدلالة على العنوان

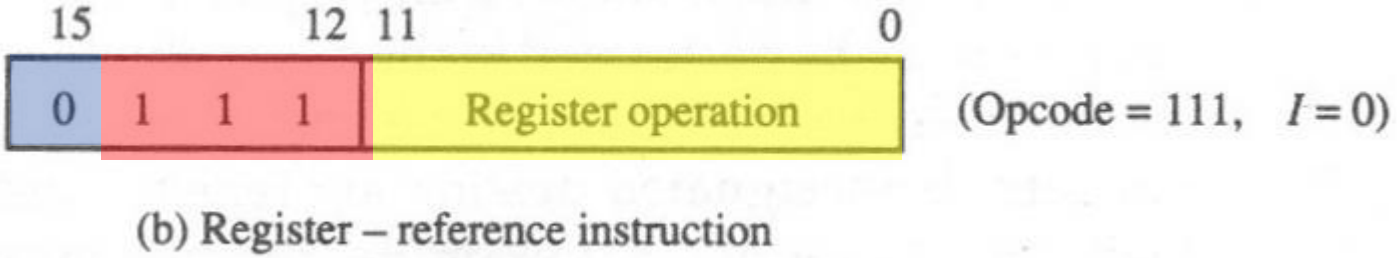
وخانة واحدة I للدلالة على نوع العنونة:

I = 0 في العنونة المباشرة

I =1 في العنونة غير المباشرة

بينما تأخذ شفرة التعليمة OP Code ثلاث خانات وتتراوح قيمتها بين 000 و 110

2- تعليمة التعامل مع مسجل Register Reference Instruction



(b) Register – reference instruction

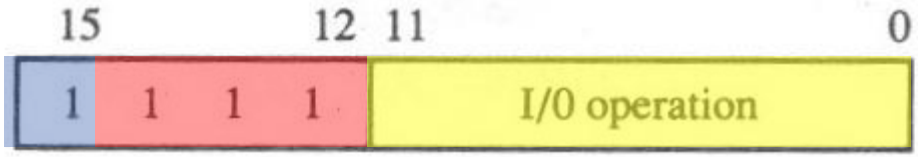
تأخذ شفرة 111 مع قيمة 0 للخانة 15 من التعليمة،

وهي تعليمة تتم مع المراكم AC أو يكون طرفا فيها،

بالتالي لا يتطلب ان تكون الذاكرة طرفا في التعليمة ولا تحتاج إلى عنوان،

لذلك تستخدم الـ 12bit الباقية لتحديد العملية أو الاختيار الذي سينفذ

3– تعليمة التعامل مع أجهزة الدخل والخرج Input Output Instruction



(Opcode = 111, $I = 1$)

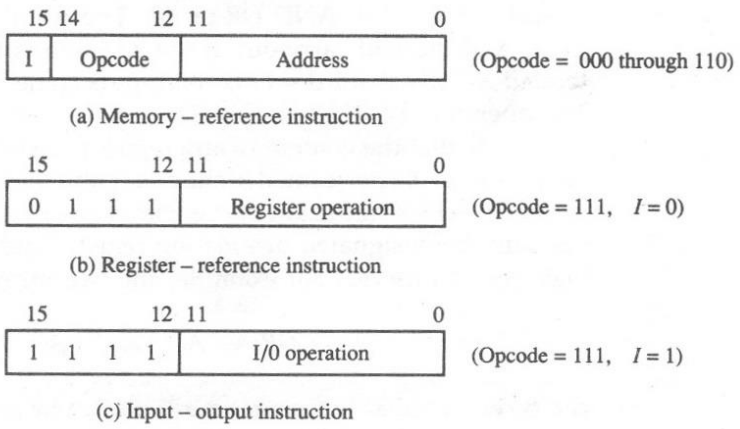
(c) Input – output instruction

وهي لا تتعامل مع الذاكرة

تكون فيها شفرة التعليمة 111 بينما تأخذ الخانة 15 قيمة 1

تستخدم الخانات الـ 12 المتبقية لتحديد نوع العملية المنجزة .

Figure 4-5 Basic computer instruction formats.



يمكن للحاسب تمييز فيما إذا كانت التعليمة تتعامل مع الذاكرة عن طريق فحص الخانات من 12 إلى 14:

- فإذا لم تكن 111 فهي تعليمة تعامل مع الذاكرة يتم بعدها تحديد نوع العنونة من الخانة I
- إذا كانت 111 عندها يمكن تحديد:

- أنها تعليمة مسجل إذا كانت الخانة 15 بقيمة 0

- وإلا فهي تعليمة دخل خرج .

طبعا" مجموع تعليمات الحاسب هي أكثر من 8 (ليس فقط كون عدد خانات شفرة التعليمة هي ثلاث من خانة 12إلى 14),

يستخدم الخانات الـ 12 (من 0 إلى 11) **في عنونة المسجل والدخل والخرج** بدلا من الدلالة على عنوان (لذلك سيتجاوز عدد التعليمات في هذا الحاسب الافتراضي الثمانية تعليمات)

بفترض أن عدد التعليمات في نظام الحاسوب الافتراضي 25 تعليمة (الحد الادنى) الجدول 2-4

تتألف كل واحدة من ثلاثة حروف تعبر عن طبيعة التعليمة وعملها وإلى يمين كل تعليمة الشفرة المناسبة بالنظام الست عشري

تدل القيمة XXX على عنوان بطول 12Bit ونلاحظ في تعليمات الذاكرة وجود شفرتين الأولى للعنونة المباشرة والثانية لغير المباشرة .

يجب على أي حاسب ان يتضمن تعليمات تؤمن للمستخدم (المبرمج) إجراء أي عمل حاسبي

تعتبر مجموعة التعليمات أنها كاملة إذا كانت تتضمن التالي:

1. تعليمات حسابية ومنطقية وتعليمات إزاحة .
2. عليمات نقل المعطيات بين الذاكرة ومسجلات الحاسب
3. عليمات تحكم وفق شروط وحالات معينة (تفريع).
4. عليمات إدخال وإخراج .

تشكل التعليمات في الجدول (2-4) تحقق بحد أدنى مجموعة تعليمات كاملة .

Table 4-2 Basic Computer Instructions

Symbol	Hexadecimal code $I = 0$	Hexadecimal code $I = 1$	Description
AND	0xxx	8xxx	AND memory word to *AC*
ADD	1xxx	9xxx	Add memory word to *AC*
LDA	2xxx	Axxx	Load memory word to *AC*
STA	3xxx	Bxxx	Store content of *AC* in memory
BUN	4xxx	Cxxx	Branch unconditionally
BSA	5xxx	Dxxx	Branch and save return address
ISZ	6xxx	Exxx	Increment and skip if zero
CLA	7800		Clear *AC*
CLE	7400		Clear *E*
CMA	7200		Complement *AC*
CME	7100		Complement *E*
CIR	7080		Circulate right *AC* and *E*
CIL	7040		Circulate left *AC* and *E*
INC	7020		Increment *AC*
SPA	7010		Skip next instruction if *AC* positive
SNA	7008		Skip next instruction if *AC* negative
SZA	7004		Skip next instruction if *AC* zero
SZE	7002		Skip next instruction if *E* is 0
HLT	7001		Halt computer
INP	F800		Input character to *AC*
OUT	F400		Output character from *AC*
SKI	F200		Skip on input flag
SKO	F100		Skip on output flag
ION	F080		Interrupt on
IOF	F040		Interrupt off

تعليمات التعامل الذاكرة

تعليمات التعامل في المسجلات

تعليمات التعامل مع الدخل والخرج

Table 4-2 Basic Computer Instructions

Symbol	Hexadecimal code $I = 0$	Hexadecimal code $I = 1$	Description
AND	0xxx	8xxx	AND memory word to *AC*
ADD	1xxx	9xxx	Add memory word to *AC*
LDA	2xxx	Axxx	Load memory word to *AC*
STA	3xxx	Bxxx	Store content of *AC* in memory
BUN	4xxx	Cxxx	Branch unconditionally
BSA	5xxx	Dxxx	Branch and save return address
ISZ	6xxx	Exxx	Increment and skip if zero
CLA	7800		Clear *AC*
CLE	7400		Clear *E*
CMA	7200		Complement *AC*
CME	7100		Complement *E*
CIR	7080		Circulate right *AC* and *E*
CIL	7040		Circulate left *AC* and *E*
INC	7020		Increment *AC*
SPA	7010		Skip next instruction if *AC* positive
SNA	7008		Skip next instruction if *AC* negative
SZA	7004		Skip next instruction if *AC* zero
SZE	7002		Skip next instruction if *E* is 0
HLT	7001		Halt computer
INP	F800		Input character to *AC*
OUT	F400		Output character from *AC*
SKI	F200		Skip on input flag
SKO	F100		Skip on output flag
ION	F080		Interrupt on
IOF	F040		Interrupt off

نلاحظ من الجدول ان مجموعة تعليمات كاملة يجب ان تحتوي على الاقل:

- توجد تعليمة حسابية هي الجمع ADD وتعليمة المتمم CMA وتعليمة زيادة واحد INC
- يمكن استخدام تلك التعليمات الثلاث لتنفيذ تعليمة الطرح باستخدام تمثيل الثنائي
- أيضا توجد تعليمي إزاحة إلى اليمين CIR وإلى اليسار CIL
- يمكن تنفيذ تعليمتي الضرب والقسمة باستخدام الجمع والطرح والإزاحة.

Table 4-2 Basic Computer Instructions

Symbol	Hexadecimal code $I = 0$	Hexadecimal code $I = 1$	Description
AND	0xxx	8xxx	AND memory word to *AC*
ADD	1xxx	9xxx	Add memory word to *AC*
LDA	2xxx	Axxx	Load memory word to *AC*
STA	3xxx	Bxxx	Store content of *AC* in memory
BUN	4xxx	Cxxx	Branch unconditionally
BSA	5xxx	Dxxx	Branch and save return address
ISZ	6xxx	Exxx	Increment and skip if zero
CLA	7800		Clear *AC*
CLE	7400		Clear *E*
CMA	7200		Complement *AC*
CME	7100		Complement *E*
CIR	7080		Circulate right *AC* and *E*
CIL	7040		Circulate left *AC* and *E*
INC	7020		Increment *AC*
SPA	7010		Skip next instruction if *AC* positive
SNA	7008		Skip next instruction if *AC* negative
SZA	7004		Skip next instruction if *AC* zero
SZE	7002		Skip next instruction if *E* is 0
HLT	7001		Halt computer
INP	F800		Input character to *AC*
OUT	F400		Output character from *AC*
SKI	F200		Skip on input flag
SKO	F100		Skip on output flag
ION	F080		Interrupt on
IOF	F040		Interrupt off

- كذالك توجد تعليمات منطقية كالمتمم CMA وتصفير المراكم CLA وتعليمة AND
- وباستخدام AND والمتمم نحصل على NAND والتي منها يمكن تنفيذ أي تابع منطقي
- كما توجد تعليمة نقل من الذاكرة إلى المراكم LDA وتعليمة تخزين المراكم في الذاكرة STA واللتان تستخدمان لنقل المعطيات
- تعطي تعليمات القفز BUN و BSA والتفريع , SKO SKI , SZA , ANAالحل المناسب لإجراء التحكم بالبرامج
- أخيرا تزمن تعليمتي الإدخال INP والإخراج OUT عملية التوصل بين الحاسب والمحيط الخارجي

4- التزامن والتحكم Timing And Control

يتم تأمين التزمن في الحاسب الافتراضي عن طريق **مولد نبضات أساسي** موصول إلى جميع المسجلات والقلابات إلى وحدة التحكم.

لا يتم تغيير حالة مسجل إلا بتطبيق نبضات عليه مع تمكين خطوط التحكم المناسبة له

يمكن توليد إشارات التحكم عن طريق برنامج مخزن في الذاكرة والذي بتغييره ونحصل على إشارات تحكم مغيرة وتدعى هذه الطريقة **بالتحكم الميكروي المبرمج Micro programmed Control**

يبين الشكل 4-6 لمخطط الصندوقي لوحدة التحكم والتي تتألف من :

- مفكك شفرة
- عداد متسلسل Counter Sequence
- مجموعة بوابات تحكم منطقية

شرح الدارة:

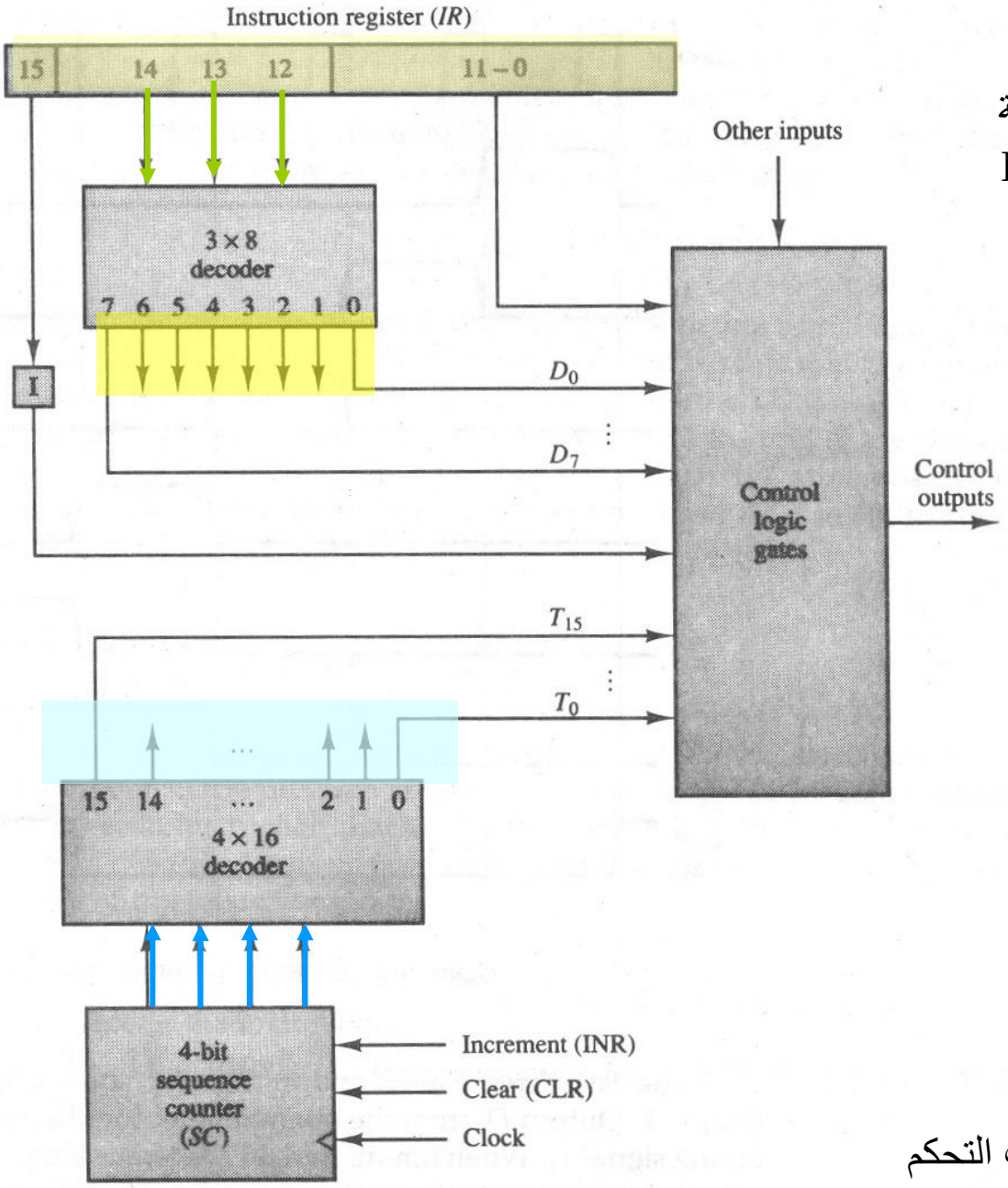


Figure 4-6 Control unit of basic computer.

تقرأ التعليمة من الذاكرة وتوضع في مسجل التعليمة IR بعدها توضع الخانة 15 في قلاب FF خاص I

أما الخانات الثلاث من 12 إلى 14 فتدخل إلى مفكك شفرة 3X8 يعطي على خرجه خطوط تحكم من D_0 إلى D_7 متوافقة من قيمة الخانات الثلاث

أما خرج العداد المتسلسل SC والذي هو عداد تصاعدي بأربع خانات فتوصل على مدخل مفكك شفرة 4X16

إشارة من T_{15} متوافقة مع قيمة العداد SC

وتوصل الخانات من 0 إلى 11 من التعليمة

وكذلك D_0 ÷ D_7

وأيضا T_0 حتى T_{15}

وكذلك خرج القلاب I إلى مداخل مجموعة بوابات التحكم المنطقية إلى مداخل أخرى

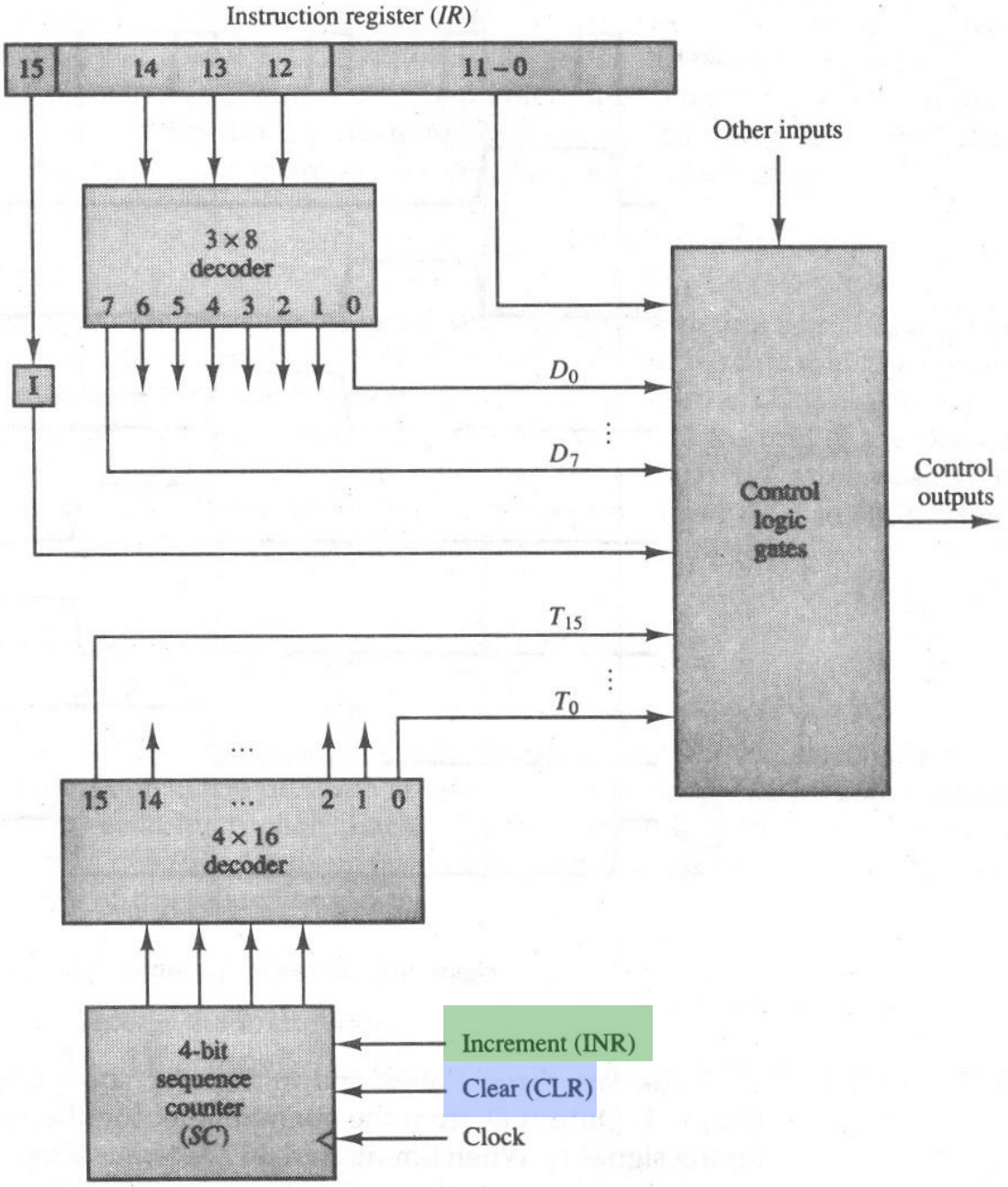


Figure 4-6 Control unit of basic computer.

يتم تصفير العداد SC باستخدام مدخل تصفير Clear (CLR)،

يبدأ العداد بالعد عند تمكين مدخل الزيادة Increment (INR) وفقاً لنبضات ساعة تؤمن له عن طريق دخل نبضات الساعة Clock

مثلا نعتبر أن العداد يتزايد ليعطي الإشارات T_0،T_1،T_2،T_3،T_4 على التتالي

وفي اللحظة T_4 يأخذ SC قيمة الصفر إذا كانت D_3 فعالة

نعبر عن هذا بالعبارة التالية :

$$D_3\,T_4 : SC \leftarrow 0$$

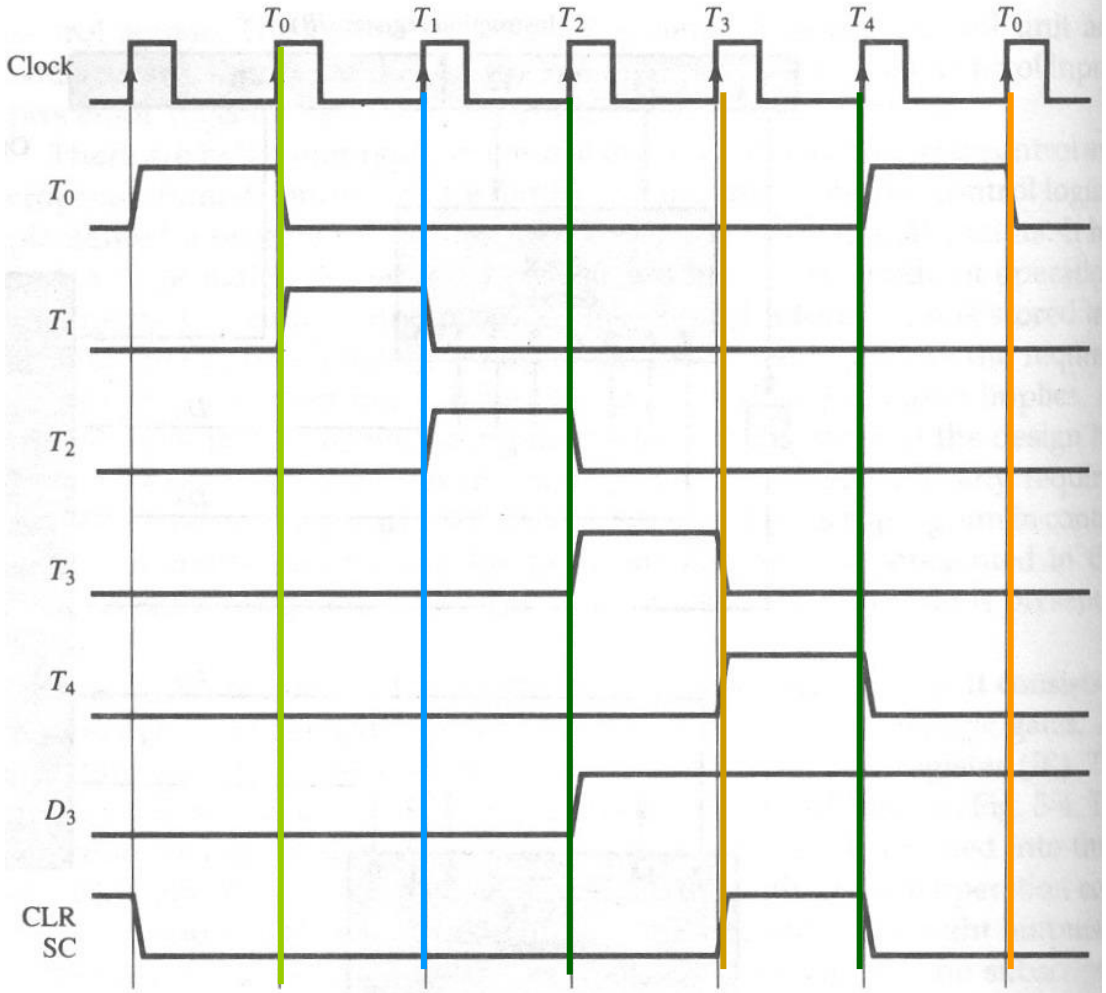


Figure 4-7 Example of control timing signals.

ان تسلسل نبضات التحكم لهذا المثال لها الشكل :

تصبح T_0 فعالة عند الجبهة الصاعدة من نبضة الساعة بعد تصفير SC
وفي النبضة الثانية T_1 فعالة
وفي الثالثة T_2
وفي الرابعة T_3
وفي الخامسة T_4

يفترض في هذه اللحظة أن D_3 فعالة (وقد تكون فعالة قبل هذه اللحظة)
عندها يتحقق الشرط D_3 T_4
وبالتالي يأخذ SC قيمة 0 (CLR) أي
تصبح T_0 فعالة من جديد عند أول جبهة صاعدة ثم T_1 وهكذا . .....

أما عند التعامل مع الذاكرة فيجب أن تنجز عملية القراءة أو الكتابة خلال نبضة ساعة واحدة

وعادة تستغرق هاتين العملتين وقتا" أطول من تردد الحاسب مما يستدعي استخدام نبضات تأخير

أما في حاسبنا الافتراضي فإننا نستخدم ترددا منخفضا بحيث يمكن أن تتم القراءة أو الكتابة في الذاكرة خلال نبضة ساعة واحدة .

وسنعتمد لدى تمثيل التعليمة على النبضات $T_0 \div T_{15}$ والتي تمثل التزامن

فمثلا سنستخدم العبارة :

$$T_0 : AR \leftarrow PC$$

تعني انه يتم تحميل محتويات عداد البرنامج PC إلى مسجل العناوين AR عند نبضة الساعة الأولى T_0 من دورة الآلة الحالية.

يتم ذلك بوضع محتويات PC على الممر المشترك بتطبيق $S_2\ S_1\ S_0 = 010$ وتمكين المدخل LD من المسجل AR

يتم النقل فعليا عند نهاية النبضة T_0 أي مع بداية انتقال SC من 0000 إلى 0001 حيث تصبح T_1 فعالة و T_0 غير فعالة

5- دورة التعليمة Instruction Cycle :

يتألف البرنامج في الذاكرة من سلسلة تعليمات، حيث يقوم الحاسب بتنفيذ البرنامج عبر دورة متكررة لكل تعليمة،

وتتضمن كل دورة دورات جزئية أو أطوار Phases.

في حاسبنا الافتراضي تتألف دورة التعليمة من الأطوار التالية .

1. **إحضار تعليمة من الذاكرة .**
2. **تشفير أو فك تشفير التعليمة .**
3. **قراءة العنوان الفعال من الذاكرة إذا كانت العنونة غير مباشرة .**
4. **تنفيذ التعليمة .**

عند الانتهاء من الخطوة الرابعة يعود التحكم إلى التنفيذ من الخطوة الأولى أي إحضار وفك تشفير وتنفيذ التعليمة التالية

وتستمر هذه العمليات بالتكرار حتى الوصول إلى تعليمة HALT

❖ إحضار وفك تشفير التعليمة Fetch And Decode:

عادة يحمّل عداد البرنامج PC بعنوان أول تعليمة من البرنامج ويأخذ العداد التسلسلي SC قيمة الصفر مولدا إشارة تزامن T_0

وبتزايد SC بمقدار 1 بعد كل نبضة ساعة مولدا إشارات تزامن T_0،T_1،T_2 ..

وبالتالي يمكن تمثيل طور جلب التعليمة وفك تشفيرها بعبارات نقل المسجل التالية :

T_0 : AR ← PC
T_1 : IR← M[AR], PC ← PC+1
T_2 : D_0.. D_7 ← Decode IR(12-14), AR ← IR(0-11), I ← IR(15)

بما أن AR هو المسجل الوحيد الموصول إلى خطوط العناوين في الذاكرة فإنه من الضروري أن يتم نقل عنوان التعليمة المراد جلبها والموجود في PC إلى AR،

يتم ذلك خلال نبضة التزامن الأولى T_0

R

T_0 : AR ← PC
T_1 : IR← M[AR], PC ← PC+1
T_2 : D_0.. D_7 ← Decode IR(12-14), AR ← IR(0-11), I ← IR(15)

بعدها تتم القراءة من الذاكرة للتعليمة ووضعها في المسجل IR مع زيادة PC بمقدار واحد للدلالة على التعليمة التالية ويتم ذلك خلال نبضة التزامن التالية T1

في النبضة T2 يتم:

- فك تشفير التعليمة بنقل الخانة 15 من IR (الذي اصبح يحتوي على التعليمة المراد تنفيذها) إلى القلاب I

- نقل الجزء الخاص بالعنوان (من الخانة 0 إلى الخانة 11) إلى مسجل العناوين AR وتنتقل الخانات من 12 إلى 14 إلى مفكك الشفرة لتوليد إشارات العمليات D_0 وحتى D_7

مع ملاحظة أن SC يتزايد عند كل نبضة ساعة تلقائيا T2,T1,T0 على التسلسل

يبين الشكل (4-8) كيف اول عمليتي نقل للمسجلات عبر الممر المشترك وذلك باستخدام بوابات OR للتحكم.

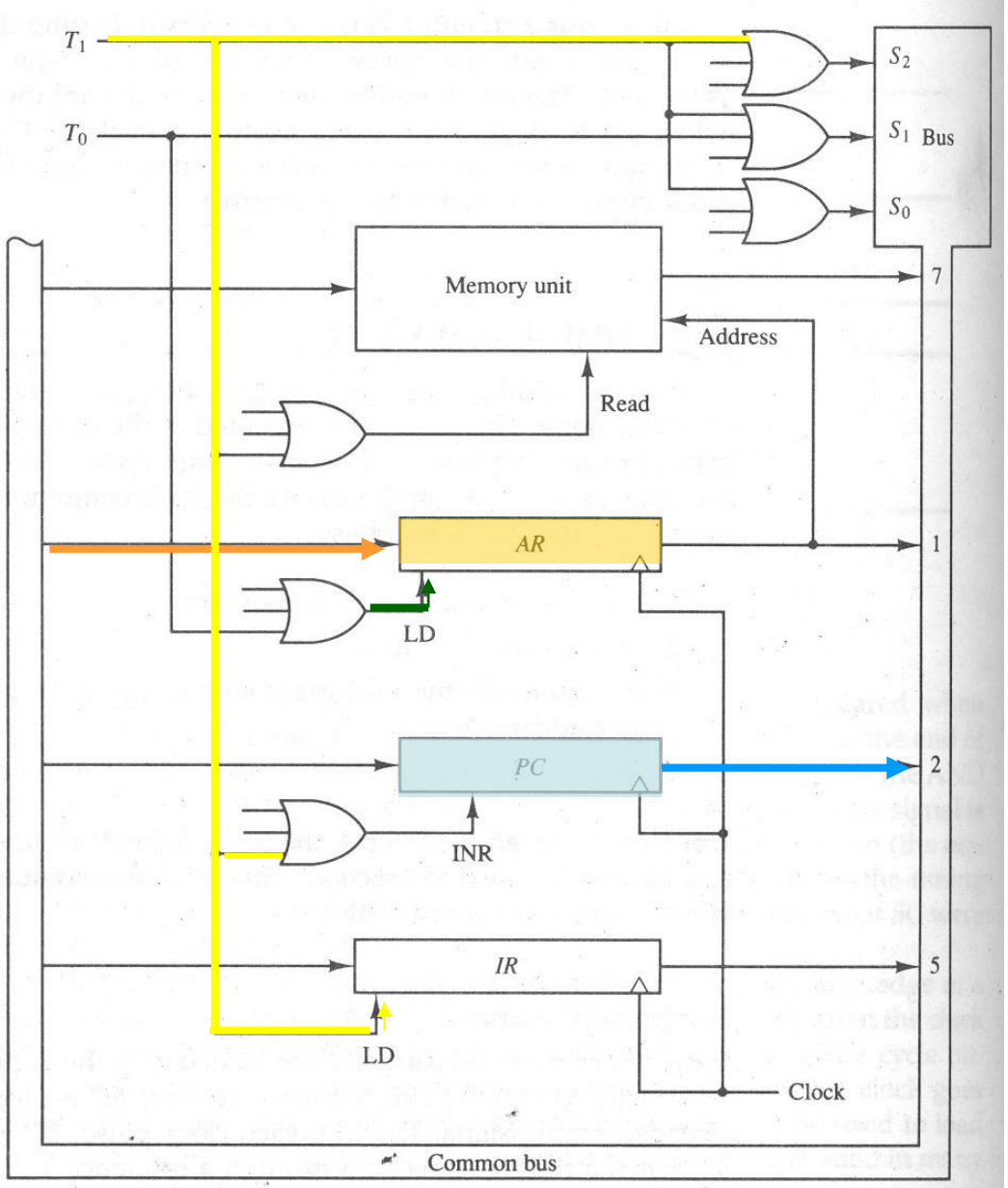


Figure 4-8 Register transfers for the fetch phase.

ولتنفيذ عملية نقل PC إلى AR يتم تنفيذ مايلي:

1. وضع محتويات PC على الممر بتطبيق $S_2 S_1 S_0 = 010$
2. نقل محتويات الممر إلى AR بتفعيل مدخل LD على AR من خلال إشارة التحكم T0

يتم ذلك عبر التزامن $T_0=1$ وبالتالي توصل T_0 بحيث تعطي $S_2S_1S_0 = 010$

كما تسمح لـ LD بان يكون 1 كالشكل .

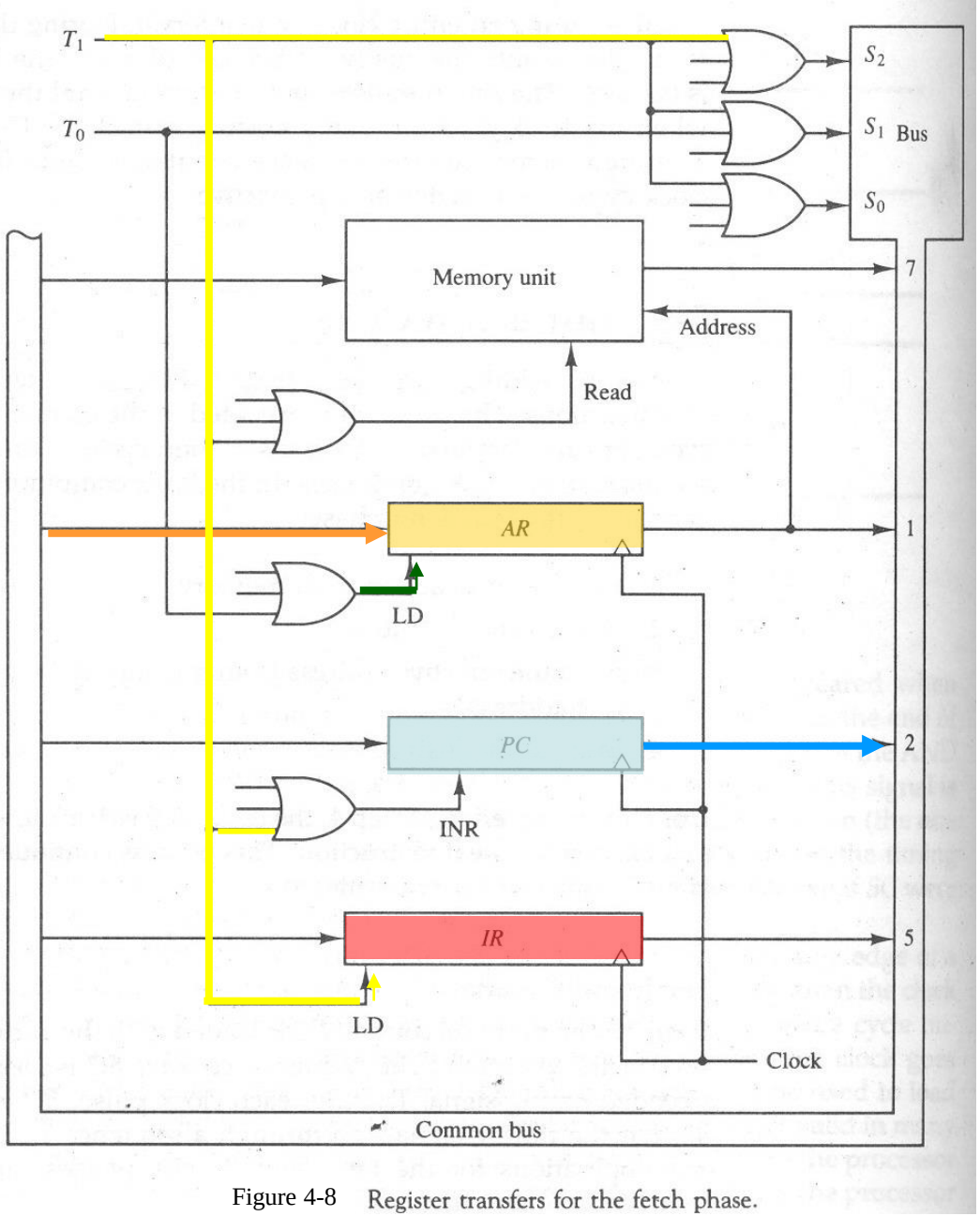


Figure 4-8 Register transfers for the fetch phase.

أما لتنفيذ العبارة الثانية

T1 : IR← M[AR], PC ← PC+1

فإنه يجب على T1 أن تقدم ما يلي:

1. تمكين مدخل القراءة Read من الذاكرة *(من خلال خط التحكم T1)*
2. وضع محتويات الممر بتطبيق 111= $S_2S_1S_0$
3. نقل محتويات الممر إلى IR بتمكين مدخل LD لديه *(من خلال إشارة التحكم T1)*
4. زيادة PC بمقدار واحد بتمكين مدخل INR للمسجلPC *(من خلال إشارة التحكم T1)*

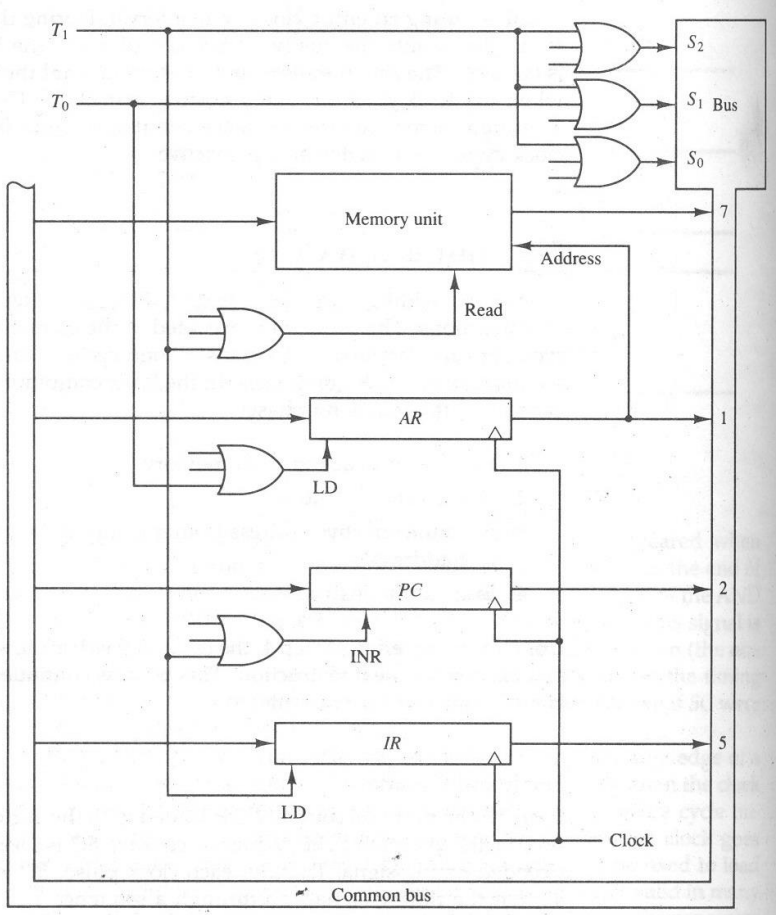


Figure 4-8 Register transfers for the fetch phase.

ونلاحظ من الشكل أن T1 يسمح بالقراءة من الذاكرة عبر تمكين Read،

يضع S2S1S0 = 111 عبر بوابات OR الثلاث ويعطي إشارة تمكين LD =1 للمسجل IR

ويعطي أيضا إشارة تمكين INR =1 للمسجل PC.

❖ تحديد نوع التعليمة Determine The Type Instruction :

يتم خلال نبضة التزامن T3 تحديد وظيفة التعليمة المقروءة من الذاكرة :

إما التعامل مع ذاكرة D7 =0
أو التعامل مع مسجل D7 =1
أو التعامل مع (I/0) D7 =1

ويبين المخطط المنهجي (4-9) دورة التعليمة وكيفية تحديد نوع التعليمة

يتم توليد D7 عبر مفكك الشفرة ذو الدخل 12 إلى 14 من التعليمة: فيكون D7 = 1 عندما تكون تلك الخانات الثلاث 111

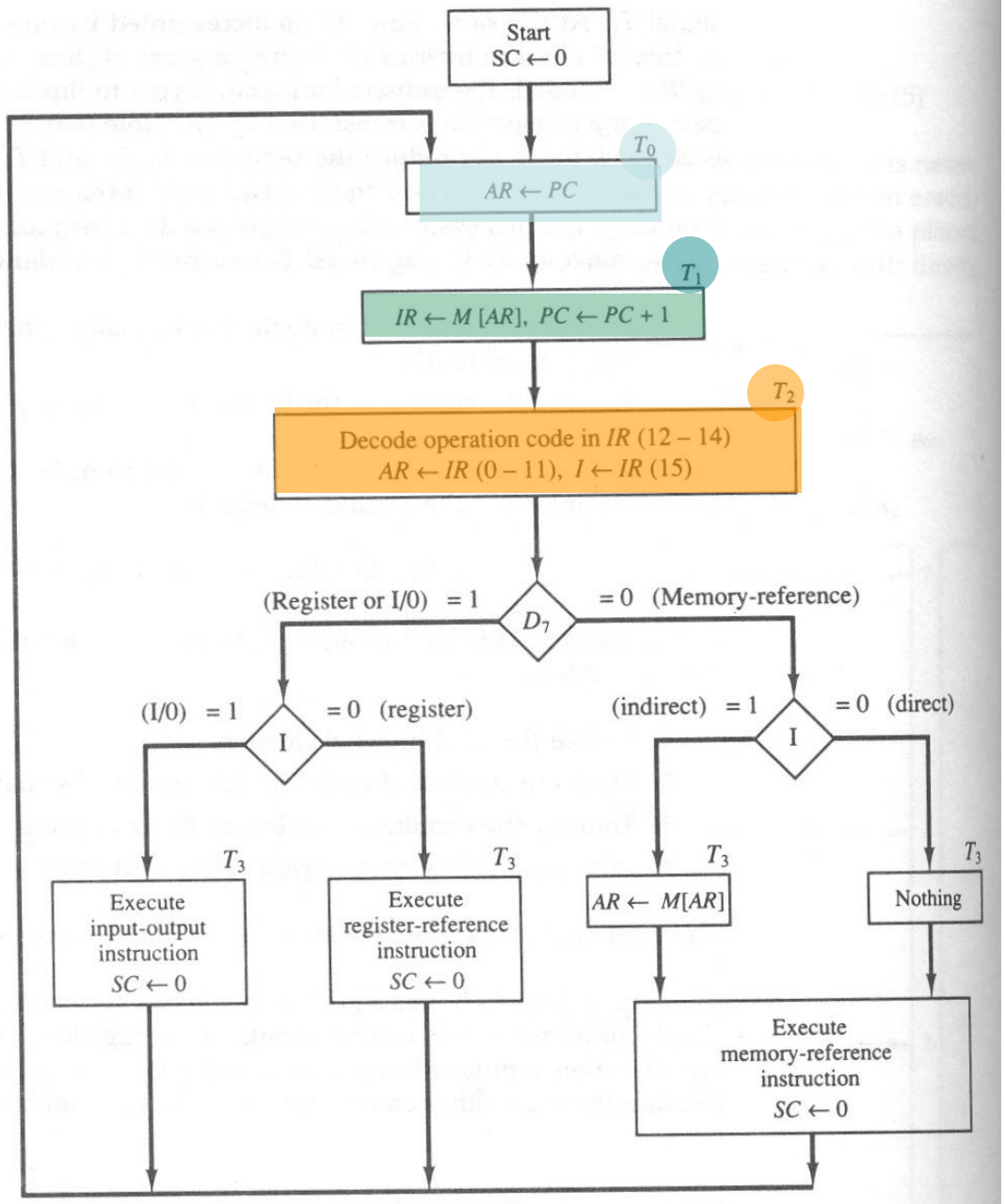


Figure 4-9 Flowchart for instruction cycle (initial configuration).

فإذا كانت $D_7 = 0$ أي خانات شفرة العمليات هي من 000 إلى 110 :

تكون التعليمة هي ذاكرة

عندها يفحص التحكم الخانة 15 من التعليمة والموضوعة في القلاب I

فإذا كان I=1 يكون لدينا تعليمة تعامل مع الذاكرة بعنونة **غير مباشرة** ومن الضروري إحضار العنوان الفعال من الذاكرة

أي تنفيذ نقل مسجل AR ← M [AR]

من خلال تفعيل مدخل Read من الذاكرة ووضع محتوياتها على الممر بتمكين خطوط الانتخاب S2 S1 S0 = 111 وتفعيل مدخل LD من المسجل AR

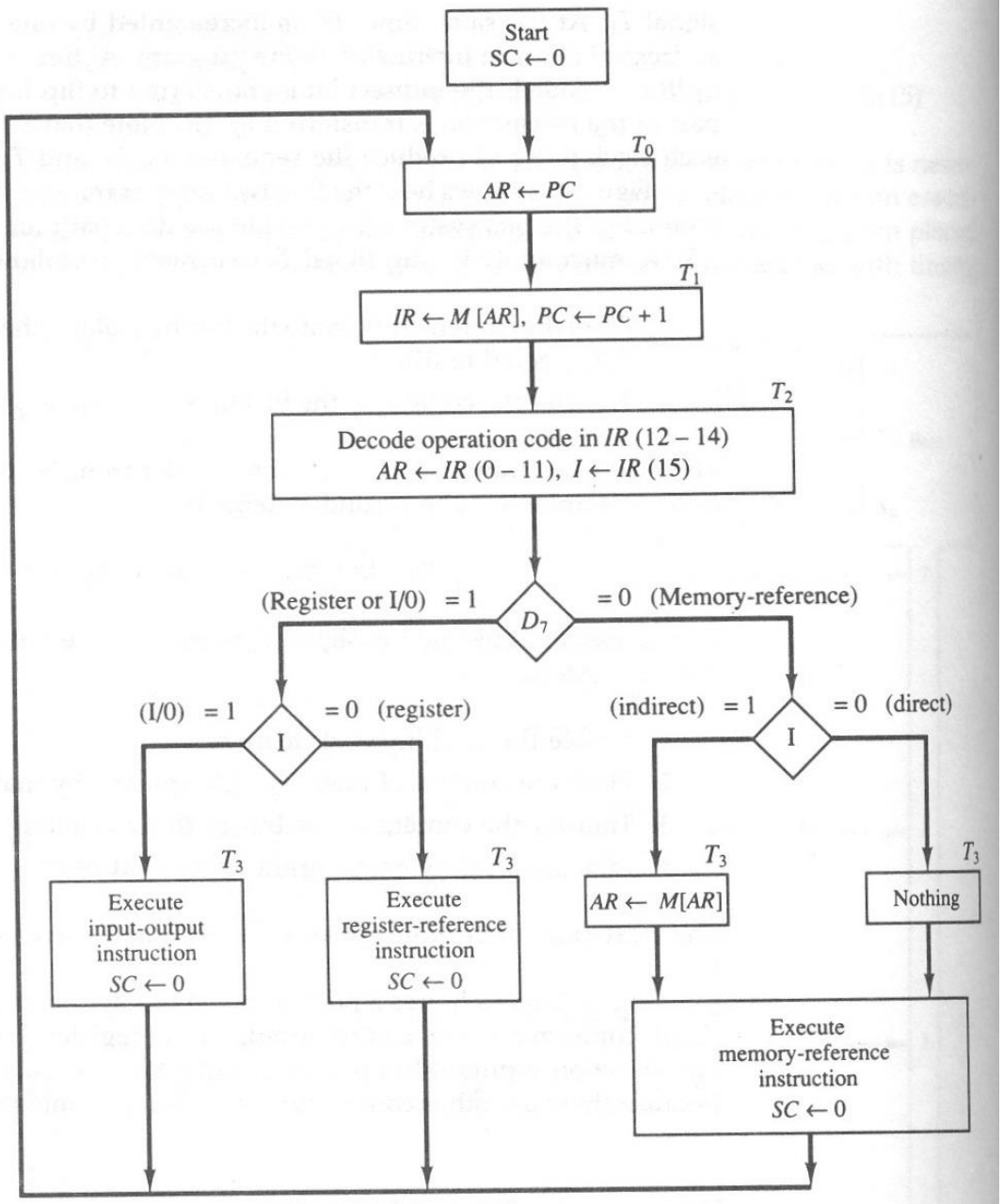


Figure 4-9 Flowchart for instruction cycle (initial configuration).

أما إذا I=0 :عندها لدينا تعليمة تعامل مع الذاكرة **بعنونة مباشرة** ولا يتم فعل شيء خلال نبضة التزامن T3

وفي كلا حالتي I فإنه يتم تنفيذ تعليمة الذاكرة بداء من النبضة T4

أما إذا كانت D7=1 فإن التحكم يعالج قيمة I

فإذا كانت I=0 فإن التعليمة خاصة بالمسجل
أما إذا كانت I=1 فالتعليمة خاصة بأجهزة الدخل والخرج

وفي كلا الحالتين يتم تنفيذ التعليمة في النبضة T3 وتصفير SC في نهاية التنفيذ لبدء دورة تعليمة جديدة ويمكن تمثيل ما سبق بالعبارات التالية :

$D'_7 IT_3$: AR ← M[AR]

$D'_7 I'T_3$: NOTHING

$D_7 I'T_3$: EXECUTE A REGISTER-REFERENCE INSTRUCTION

$D_7 IT_3$: EXECUTE AN INPUT -OUTPUT INSTRUCTION

D_7	I	T_3	Operation
0	1	1	تعامل مع ذاكرة وعنونة غير مباشرة
0	0	1	تعامل مع ذاكرة وعنونة مباشرة
1	0	1	تعامل مع مسجل
1	1	1	تعامل مع دخل /خرج

❖ تعليمات المسجل Register-Reference Instructions :

يتم تمييز تعليمة المسجل عندما يكون D7=1 و I=0

عندها يمكن تحديد تعليمة مسجل عن أخرى باستخدام الخانات من 0 وحتى 11 من التعليمة والموجودة في AR

يبين الجدول (3-4) مجموعة تعليمات المسجل وطريقة تنفيذها مع تمثل شرط تنفيذ $D_7I`T_3$ بالرمز r والخانة الفعالة بالرمز B_i وهذه التعليمات هي :

CLA: تصفير المسجل AC
CLE:تصفير علم الحمل E
CMA : وضع متمم المراكم الأحادي في AC (NOT)
CME: إتمام العلم E
CIR: إزاحة المراكم دائريا نحنو اليمين .
CIL: إزاحة المراكم دائريا نحو اليسار
INC: زيادة واحد إلى المراكم .
SPA: تجاوز التعليمة التالية إذا كان المراكم موجبا (أي 0= (AC (15)
SNA: تجاوز التعليمة التالية إذا كان المراكم سالبا (أي 1 = (AC (15)
SZA: تجاوز التعليمة التالية إذا كان المراكم يساوي الصفر .
SZE: تجاوز التعليمة التالية إذا كان علم الحمل E يساوي الصفر (لا يوجد حمل أو استعارة)
HLT: إيقاف الحاسب حيث يتم وضع القلاب S في حالة الصفر وهو قلاب يشير إلى استمرار تنفيذ التعليمات في الحاسب (Start Stop Flip-FlOP) .

وفي جميع هذه التعليمات يوضع العداد المتتالي SC بقيمة صفر لبدء دورة تعليمة جديدة .

Table 4-3 Execution of Register-Reference Instructions

$D_7I'T_3 = r$ (common to all register-reference instructions)
$IR(i) = B_i$ [bit in $IR(0–11)$ that specifies the operation]

	r:	$SC \leftarrow 0$	Clear SC
CLA	rB_{11}:	$AC \leftarrow 0$	Clear AC
CLE	rB_{10}:	$E \leftarrow 0$	Clear E
CMA	rB_9:	$AC \leftarrow \overline{AC}$	Complement AC
CME	rB_8:	$E \leftarrow \overline{E}$	Complement E
CIR	rB_7:	$AC \leftarrow \text{shr } AC, AC(15) \leftarrow E, E \leftarrow AC(0)$	Circulate right
CIL	rB_6:	$AC \leftarrow \text{shl } AC, AC(0) \leftarrow E, E \leftarrow AC(15)$	Circulate left
INC	rB_5:	$AC \leftarrow AC + 1$	Increment AC
SPA	rB_4:	If $(AC(15) = 0)$ then $(PC \leftarrow PC + 1)$	Skip if positive
SNA	rB_3:	If $(AC(15) = 1)$ then $(PC \leftarrow PC + 1)$	Skip if negative
SZA	rB_2:	If $(AC = 0)$ then $PC \leftarrow PC + 1)$	Skip if AC zero
SZE	rB_1:	If $(E = 0)$ then $(PC \leftarrow PC + 1)$	Skip if E zero
HLT	rB_0:	$S \leftarrow 0$ (S is a start–stop flip-flop)	Halt computer

6- تعليمات الذاكرة Memory-Reference Instructions:

يتم تنفيذ تعليمات الذاكرة بداء من النبضة T4 ويحقق الشرط D7=0 أي أن شفرة العمليات تتراوح بين D0 و D6 وبالتالي توجد لدينا سبع تعليمات (3 خانات تحدد 8 احتمالات وإشغال حالة 111 وبالتالي يبقى 7 احتمالات) نحددها كما يلي :

AND:

تنفيذ عندها D0= 1 وتقوم بإجراء عملية AND منطقية بين المراكم وحجرة الذاكرة المعينة في التعليمة ووضع ناتج العملية في المراكم

وبما أن وحدة المنطق تقبل العملية بين المراكم والمسجل DR فإنه يتم نقل معامل الذاكرة إلى إلـ DR أولا خلال النبضة T4 ومن ثم إجراء AND بين المراكم و DR ووضع الناتج في المراكم وتصفير SC في النبضة T5أي:

D_0T_4: DR ← M[AR]

D_0T_5 :AC ← AC^ DR , SC ← 0

(تحتاج نبضتين)

ADD

تنفذ عندما D1 =1 وتقوم بإجراء جمع حسابي بين المراكم وحجرة الذاكرة المعينة مع وضع الحمل Cout في القلاب E, وبإجراء نفس المناقشة السابقة نجد:

D_1T_4 : DR ← M[AR]

D_1T_5 : AC ← AC + DR, E ← C_{out} , SC ← 0

(تحتاج نبضتين)

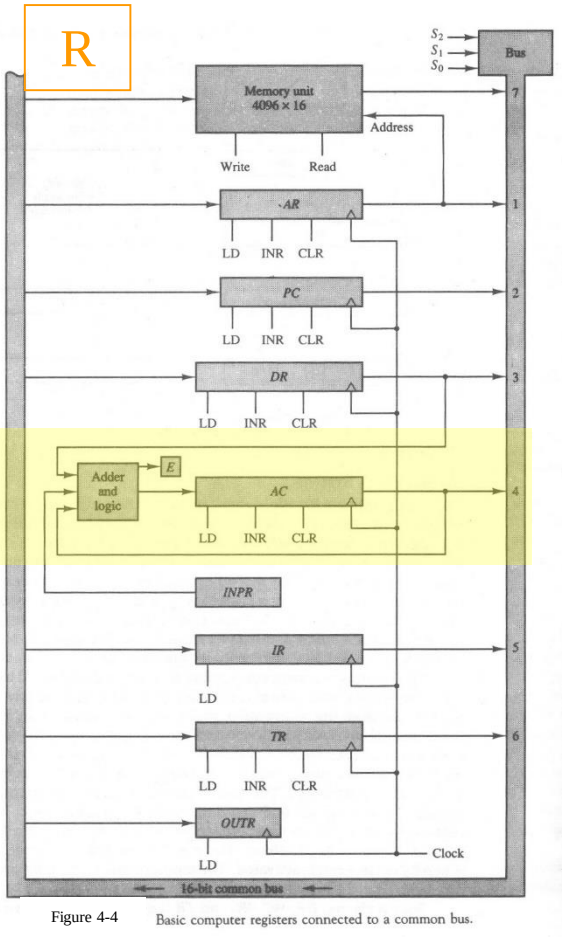


Figure 4-4 Basic computer registers connected to a common bus.

LDA

نقل محتويات الذاكرة إلى المراكم ،

ونظرا لأن المراكم غير موصول على الممر كدخل وإنما كخرج إلى الممر فقط لذلك يتم تنفيذ هذه التعليمة كالتالي :

D_2 T_4 : DR ← M[AR]

D_2T_5: AC ← DR, SC ← 0

(تحتاج نبضتين)

STA

نقل محتويات المراكم إلى الذاكرة أي :

D_3T_4 : M[AR] ← AC, SC ← 0

(تحتاج نبضة)

BUN

قفز غير مشروط إلى العنوان المعين في التعليمة أي

D_4T_4 : PC ← AR, SC ← 0

(تحتاج نبضة)

BSA

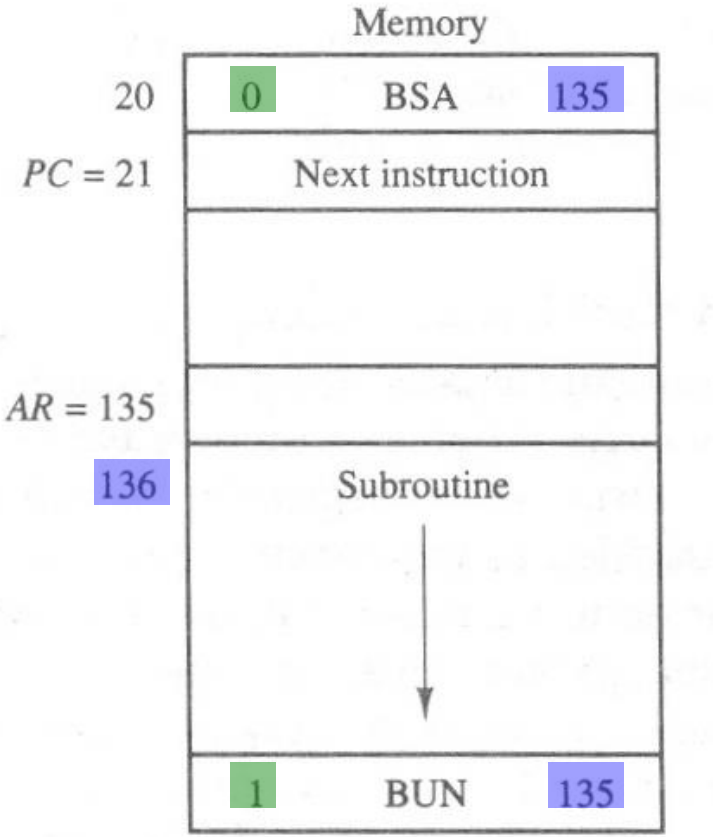
القفز مع حفظ عنوان العودة وتستخدم لمناداة برنامج فرعي ويتم تنفيذها كالتالي

D_5T_4 : M[AR] ← PC, AR ← AR + 1
D_5T_5 : PC ← AR, SC ← 0

(تحتاج نبضتين)

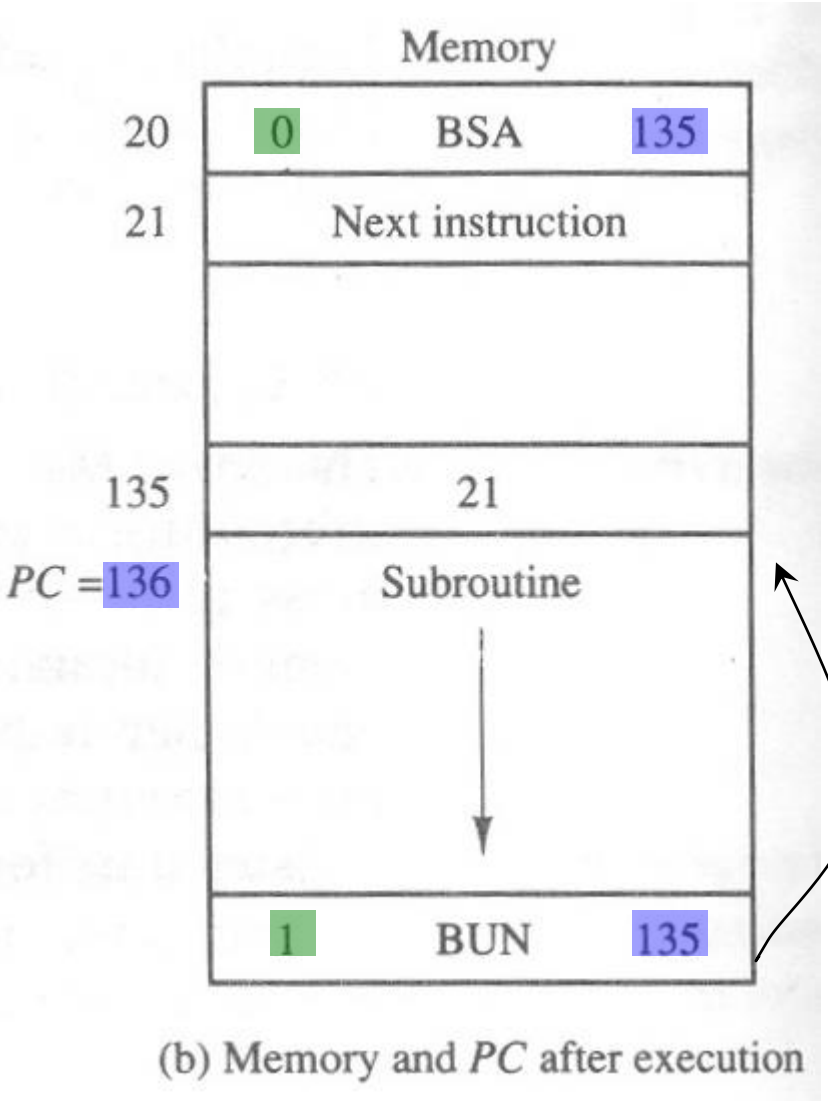
مثال :يبين الشكل 10-4 مثال على هذه التعليمة : بفرض أن الحاسب يقوم بتنفيذ تعليمة BSA بعنونة مباشرة I=0 إلى العنوان 135 وبفرض أن تعليمة القفز هذه موجودة في العنوان 20

يتحتم أن يبدأ برنامج الخدمة من العنوان 136 الذي هو 135+1 وأن ينتهي بتعليمة قفز غير مشروط BUN بعنونة غير مباشرة إلى نفس العنوان 135 الشكل (a)



(a) Memory, *PC*, and *AR* at time T_4

عندها يقوم الحاسب بوضع عنوان التعليمة التالية 21 في العنوان 135 ويبدأ التنفيذ من العنوان 136 وعند نهاية البرنامج الفرعي يتم اللجوء إلى العنوان 135 والذي يحتوي على 21 والقفز إلى هذا العنوان أي لعودة إلى 21 (غير مباشرة) كما يظهر في الشكل (b) .



(b) Memory and *PC* after execution

عند وردد التعليمة BSA يقفز على العنوان المحدد يحفظ فيه عنوان العودة

قفز لعنوان يحوي عنوان العودة

ISZ:

زيادة واحد إلى الذاكرة المعينة وتجاوز التعليمة التالية إذا كان ناتج الزيادة مساويا للصفر .أي :

D_6T_4 : DR ← M[AR]
D_6T_5 : DR ← DR + 1
D_6T_6 : M[AR] ← DR, IF DR = 0 Then PC ← PC + 1, SC ← 0

(تحتاج ثلاث نبضات)

يبين الجدول (4-4) اختصار للتعليمات السابقة ،

Table 4-4 Memory-Reference Instructions

Symbol	Operation decoder	Symbolic description
AND	D_0	$AC \leftarrow AC \wedge M[AR]$
ADD	D_1	$AC \leftarrow AC + M[AR], \quad E \leftarrow C_{out}$
LDA	D_2	$AC \leftarrow M[AR]$
STA	D_3	$M[AR] \leftarrow AC$
BUN	D_4	$PC \leftarrow AR$
BSA	D_5	$M[AR] \leftarrow PC, \quad PC \leftarrow AR + 1$
ISZ	D_6	$M[AR] \leftarrow M[AR] + 1,$ If $M[AR] + 1 = 0$ then $PC \leftarrow PC + 1$

تنفذ التعليمة المقابلة لها عندما تكون هي الفعالة

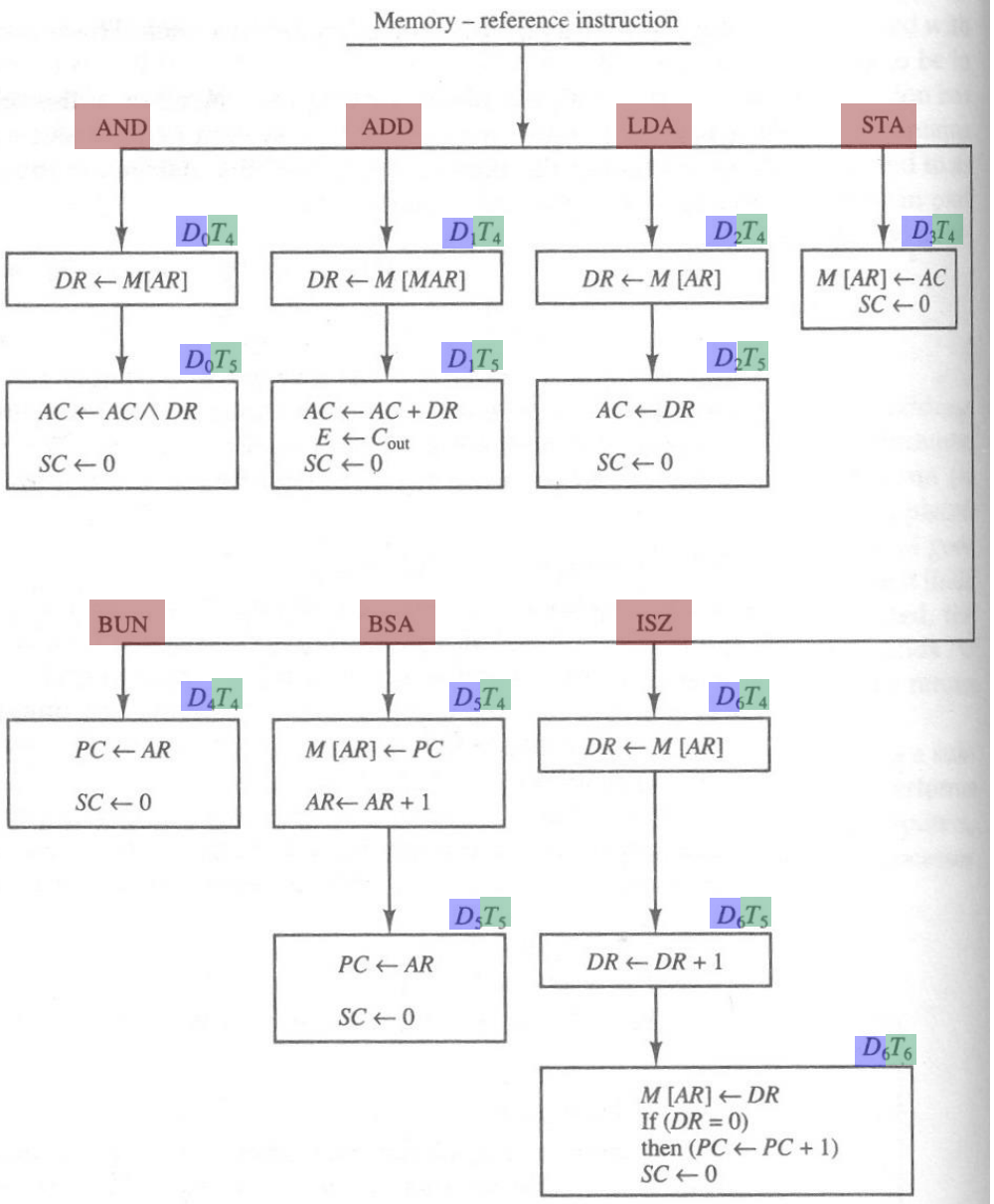


Figure 4-11 Flowchart for memory-reference instructions.

الشكل (4-11) يبين المخطط المنهجي لها والذي يبين بشكل تفصيلي أداء كل تعليمة خلال تعليمة خلال نبضات التزامن.

تكون تعليمة التعامل مع الذاكرة عندما يكون d7=0 تحدد نوع هذه التعليمة بالاعتماد على شارات التحكم $D_0 \div D_6$ وبالتالي لدينا سبع تعليمات للتعامل مع الذاكرة

علماً أن :($D_6 \div D_6$) لها القيمة ممثلة للـ Opcode وهي **000** ← 110 والممثلة للخانات 14 ÷ 12

تلفظ: تنفذ التعليمة BUN مثلا :عندما يكون D المقابلة لها وهي D4 هي الفعالة (D4=1) وتقوم بتنفيذ مايلي :

7- تعليمات الدخل والخرج والمقاطعة Input-Output And Interrupt Instructions:

تنعدم الفائدة من الحاسب ما لم يحدث اتصال بينه وبين البيئة الخارجية، فيجب أن يتم تزويد التعليمات والمعطيات المخزنة في الذاكرة من قبل جهاز دخل كما يجب أن يتم إرسال نتائج الحسابات إلى المستخدم عن طريق جهاز إخراج .

تعتبر لوحة المفاتيح Keyboard من اهم وأشهر اجهزة الدخل والطابعة printer كجهاز خرج

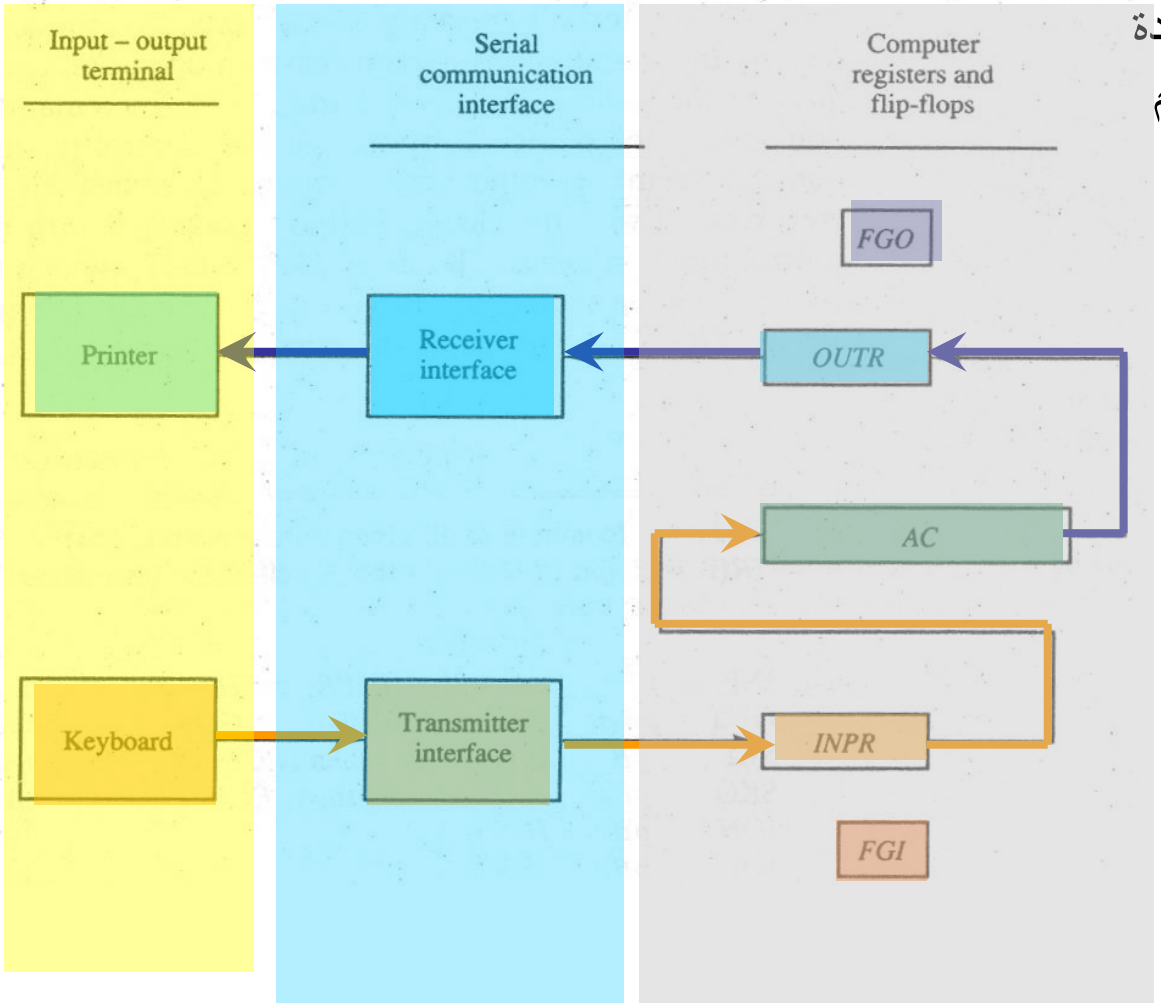
– توصيف الإدخال والإخراج Input-Output Configuration

يتم إرسال واستقبال المعطيات تسلسليا، وتتضمن كل كمية من المعطيات ثمانية خانات تشكل شفرة أبجدية رقمية Alphanumeric (**محرف**)

تتحول المعطيات إلى تفرعية عبر مسجل إزاحة تسلسلي تفرعي وبالعكس حيث تدخل المعطيات من لوحة المفاتيح إلى مسجل INPR كما تخرج المعطيات الموجودة في مسجل الخرج OUTR إلى الطابعة .

ولزيادة عملية معالجة الإدخال يتم الابتعاد عن طريقة المسح التي تستغرق وقتا طويلا عن طريق دعم مسجلي الإدخال والإخراج بقلابين الأول للإدخال FGI والثاني للإخراج FGO

Figure 4-12 Input-output configuration.



يأخذ FGI قيمة 1 عند وجود معطيات جديدة في مسجل الدخل عندها تحدث مقاطعة تقوم بقراءة محتويات INPR إلى AC وبعد قراءتها

يتم تصفير القلاب FGI أما FGO فيأخذ قيمة 1 عندما يكون OUTR قد أرسل محتوياته إلى الطابعة، وبالتالي يمكن للمسجل AC أن يرسل قيمة جديدة إليه وحالما يضع المراكم قيمة جديدة في OUTR يصبح FGO =0

ويبقى كذلك حتى إرسال هذه القيمة إلى الطابعة فيعود ليأخذ قيمة 1 معلنا جاهزيته لإرسال حرف جديد إلى الطابعة .

وتتحول المعطيات إلى تفرعية عبر مسجل إزاحة تسلسلي تفرعي وبالعكس حيث تدخل المعطيات من لوحة المفاتيح إلى مسجل INPR كما تخرج المعطيات الموجودة في مسجل الخرج OUTR إلى الطابعة.

• تعليمات الإدخال والإخراج Input-Output Instruction:

يبين الجدول (4-5) تعليمات الإدخال والإخراج وهي:

INP

إدخال محرف من المسجل INPR إلى المراكم AC وبما أن INPR بطول 8 Bit فإن المحرف ينتقل إلى الخانات 7 ÷ 0 من AC أي (0-7) AC مع وضع 0 في FGI *(لان INPR أصبح فارغا")*

OUT

إخراج محرف من المراكم إلى المسجل OUTR أي (0-7) AC ← 0.OUTR مع وضع 0 في القلاب FGO *(لأن OUTR أصبح ممتلئ)*

SKI

تجاوز التعليمة التالية في حال FG1=1 أي

If (FGI =1) then PC ← PC+1

SKO

تجاوز التعليمة التالية في حال FGO=1 أي:

If (FGO=1) Then PC ←PC + 1

ION

السماح بالمقاطعة ، وضع فيه 1 في قلاب المقاطعة IEN نقوم بالسماح بالمقاطعة.

IEN ← 1

IOF

عدم السماح بالمقاطعة .

IEN ← 0

يتم تنفيذ التعليمات السابقة عند نبضة التزامن T3 وهي تعليمات دخل /خرج

وبالتالي يجب أن تكون شفرة التعليمة 111أي (D7=1) كما يجب ان تكون I=1

إذا تحدث التعليمات السابقة عندما D7IT3 وسنرمز لها بالرمز P

أما Bi تمثل فعالية إحدى الخانات (11– 6)IR للحصول على شفرة التعليمة .

تذكرة ومراجعة

1- تعليمات التعامل مع المسجل تنفذ بدء من النبضة T3

2- تعليمات التعامل مع الذاكرة تنفذ بدء من النبضة T4

3- تعليمات I/0 تنفذ بدء من النبضة T3

Table 4-5 Input-Output Instructions

$D_7IT_3 = p$ (common to all input–output instructions)
$IR(i) = B_i$ [bit in $IR(6–11)$ that specifies the instruction]

	p:	$SC \leftarrow 0$	Clear SC
INP	pB_{11}:	$AC(0–7) \leftarrow INPR$, $FGI \leftarrow 0$	Input character
OUT	pB_{10}:	$OUTR \leftarrow AC(0–7)$, $FGO \leftarrow 0$	Output character
SKI	pB_9:	If $(FGI = 1)$ then $(PC \leftarrow PC + 1)$	Skip on input flag
SKO	pB_8:	If $(FGO = 1)$ then $(PC \leftarrow PC + 1)$	Skip on output flag
ION	pB_7:	$IEN \leftarrow 1$	Interrupt enable on
IOF	pB_6:	$IEN \leftarrow 0$	Interrupt enable off

(تجاوز التعليمة الحالية)

مقاطعة البرنامج Program Interrupt:

لزيادة السرعة يتم استخدام المقاطعة في عمليتي الإدخال والإخراج

يبين الشكل (4-13) المخطط الصندوقي لدورة المقاطعة حيث تم تطوير دورة جلب وتنفيذ تعليمة ليتضمنها دورة تخديم مقاطعة دخل /خرج

نعتمد قلابا مساعدا للمقاطعة نرمز له R يأخذ قيمة 0 في بداية الاقلاع نلاحظ من المخطط الصندوقي أنه إذا كانت R = 0 يتم دورة تعليمة Instruction Cycle يتم خلالها جلب التعليمة وفك تشفيرها ومن ثم إجراء عمليتين على التوازي الأولى تنفيذ التعليمة والثانية فحص علم السماح بالمقاطعة فإذا كان 0= IEN فيتم العودة مع بقاء R=0 وبالتالي تنفيذ دورة تعليمة جديدة

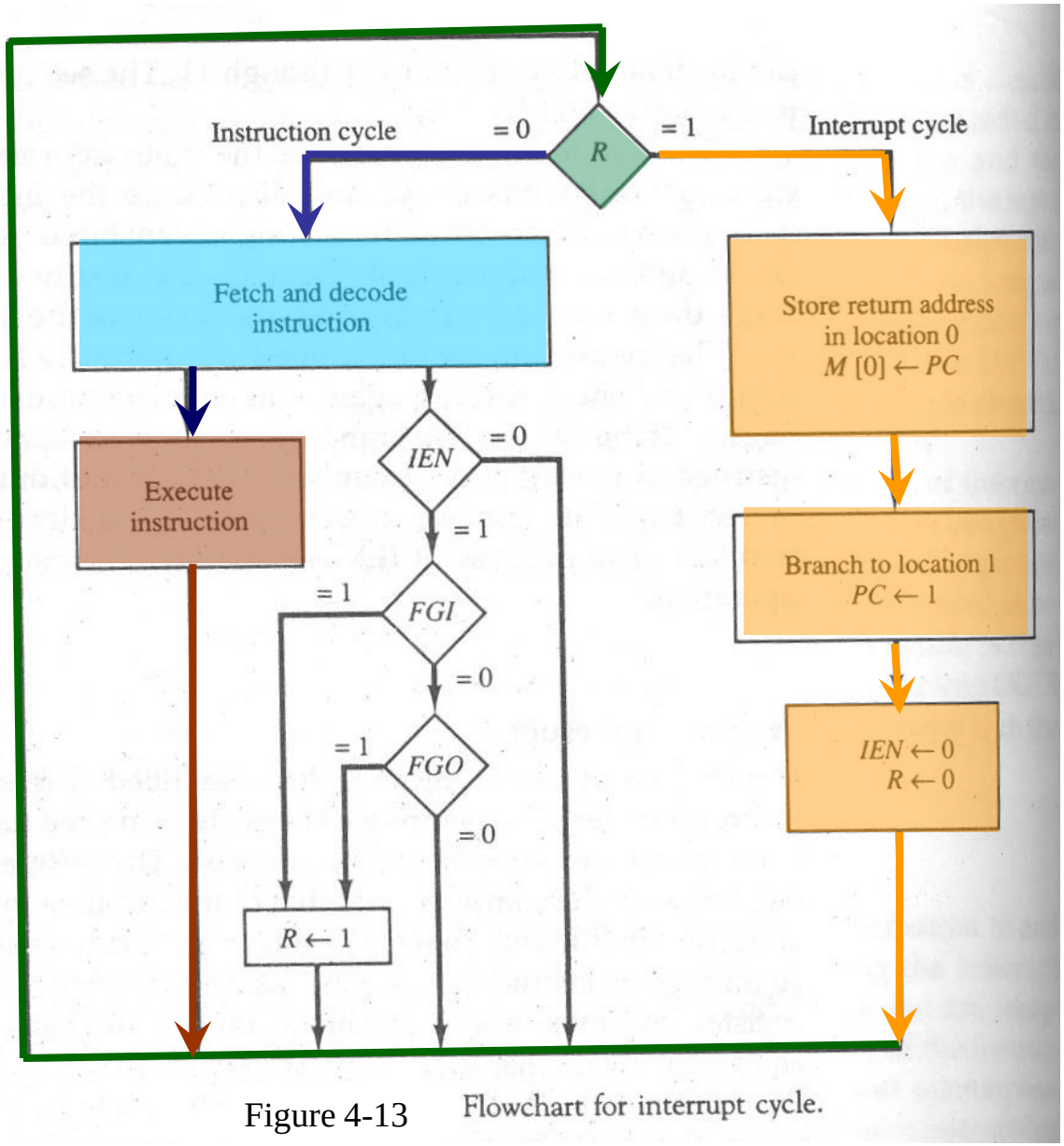


Figure 4-13 Flowchart for interrupt cycle.

أما إذا كان IEN =1 (سماح بالمقاطعة) فيتم فحص علم الدخل FGI وعلم الخرج FGO

فإذا كان أحدهما يساوي 1 يتم وضع R=1 لتنفيذ دورة مقاطعة وإلا يبقى R=0 لتنفيذ دورة تعليمة .

وعند العودة إلى البداية وبقيمة R=1 يتم تنفيذ دورة مقاطعة Cycle Interrupt والتي تتضمن حفظ عنوان العودة في الذاكرة ذات العنوان 0 أي

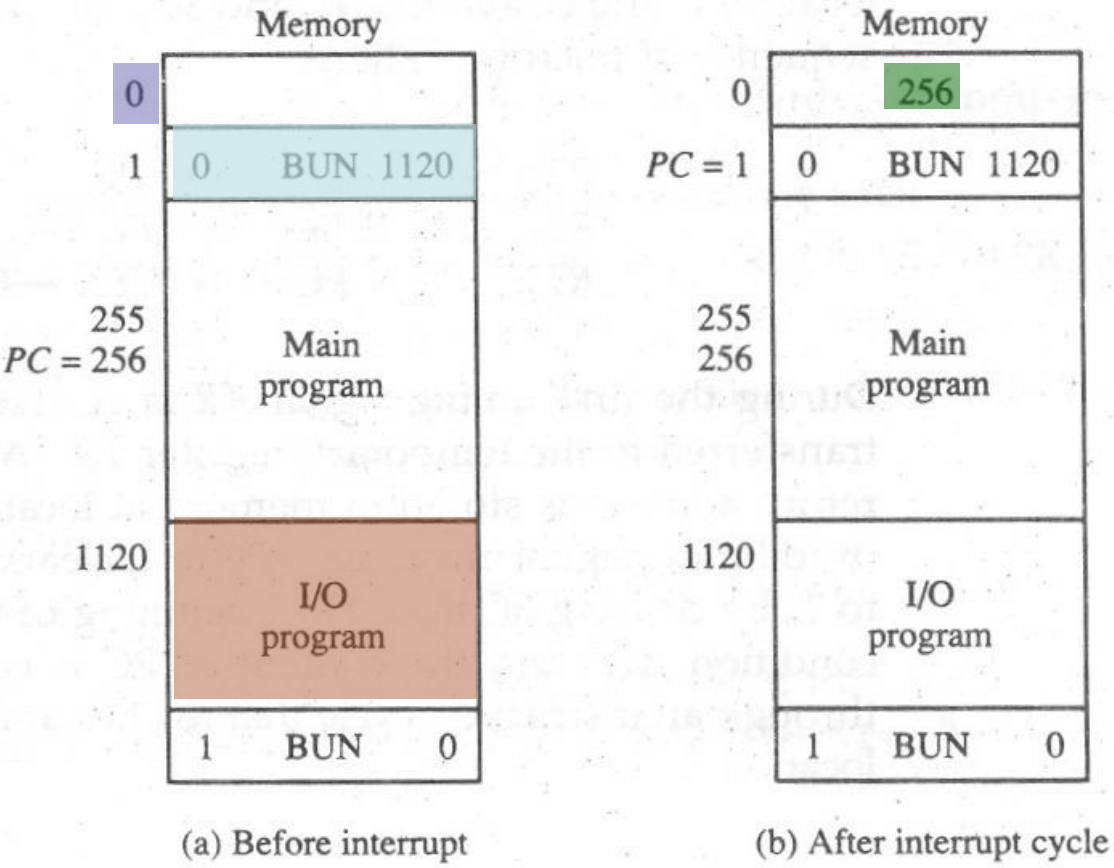
M[0] ← PC

والقفز إلى العنوان 1 أي 1 ← PC مع تصفير السماح بالمقاطعة IEN =0

وكذلك R=0 والعودة لتنفيذ برنامج خدمة المقاطعة ومع بقاء IEN= 0 يتم تنفيذ برنامج الخدمة كاملا بدون قبول مقاطعات جديدة وعند انتهاء برنامج الخدمة يتم العودة عن طريق استرداد العنوان الموجود في M[0]واستمرار التنفيذ.

مثال : على تنفيذ المقاطعة الشكل (4-14)

Fig Figure 4-14 monstration of the interrupt cycle.



بفرض برنامج رئيسي ينفذ،

عند حدوث مقاطعة لدى تنفيذ التعليمة الموجودة في العنوان 255، ينهي الحاسب تنفيذ التعليمة وينتقل إلى العنوان 0 ويضع فيه عنوان تعليمة العودة والذي هو 256 ثم ينفذ ابتداء من العنوان 1،

بفرض أن برنامج خدمة مقاطعة الدخل والخرج I/O Program يبدأ من العنوان 1120

عندها نضع في العنوان 1 تعليمة BUN 1120 وبعنونة غير مباشرة (I=0)،

عند نهاية الخدمة نضع تعليمة BUN 0 وبعنونة غير مباشرة (I=1) ليتم تنفيذ تعليمة عودة .

يمكن تلخيص :

1- تنفيذ البرنامج الرئيسي حتى طلب المقاطعة

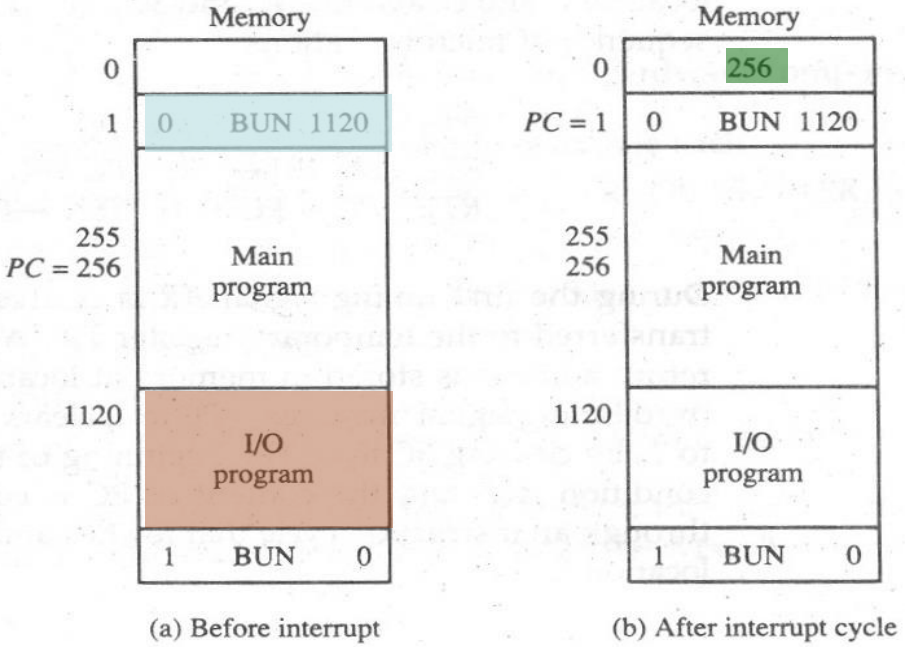
2- قفز لموقع صفر لتخزين عنوان العودة فيه

3- الانتقال إلى الموقع (1) حيث فيه عنوان بداية برنامج خدمة المقاطعة فيقفز إليه

4- تنفيذ برنامج خدمة المقاطعة المنتهي بتعليمة قفز غير مشروط إلى الموقع 0 للحصول على عنون العودة

5- الانتقال إلى عنوان العودة والمقاطعة في البرنامج الرئيسي

Figure 4-14 onstration of the interrupt cycle.



يمكن ملاحظة من الشكل (4-13) أنه يتم فحص الأعلام FGO , FGI , IEN بعد دورة إحضار التعليمة بعد مرور نبضات التزامن الثلاث T2,T1,T0 وبالتالي يمكن الحصول على عبارة شرط وضع 1 في R كالتالي

$$T`_0T`_1T`_2\ (IEN)(FGI + FGO) : R \leftarrow 1$$

اعتمادا" على التعديل الاخير انفس الشكل يمكن دمج وضعية R في المعادلات جلب التعليمة لتصبح معادلات تنفيذ دورة المقاطعة كالتالي :

RT_0 : AR←0 , TR←PC
RT_1 : M[AR]←TR , PC←0
RT_2 : PC←PC←+1 , IEN ←0 , R←0 , SC←0

8- التوصيف الكامل للحاسب الافتراضي Complete Computer Description :

بالعودة إلى التعديل الأخير والذي يضم إضافة دورة مقاطعة إلى الحاسب يصبح مخطط عمل الحاسب الافتراضي كالشكل المبين في (4-15)

R

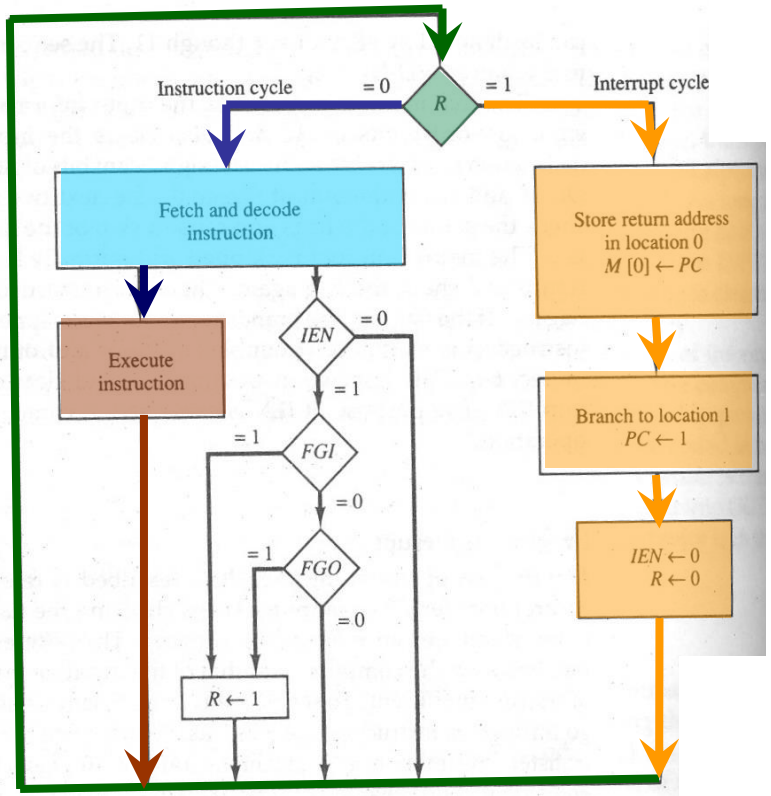


Figure 4-13 wchart for interrupt cycle.

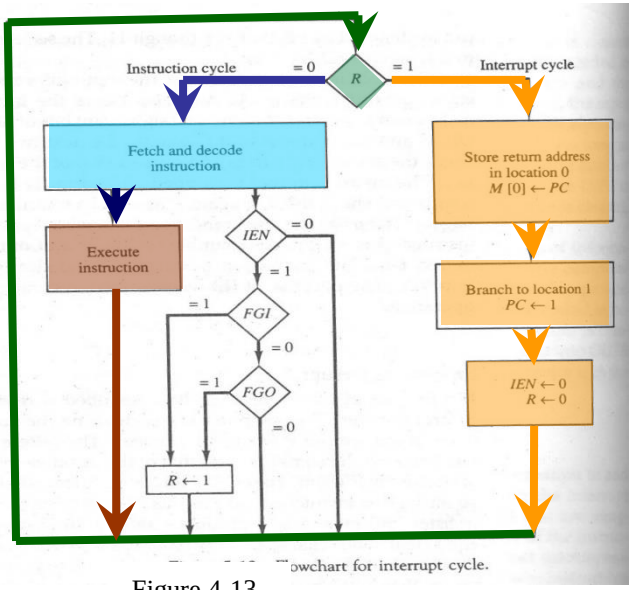


Figure 4-13

Figure 4-15 Flowchart for computer operation.

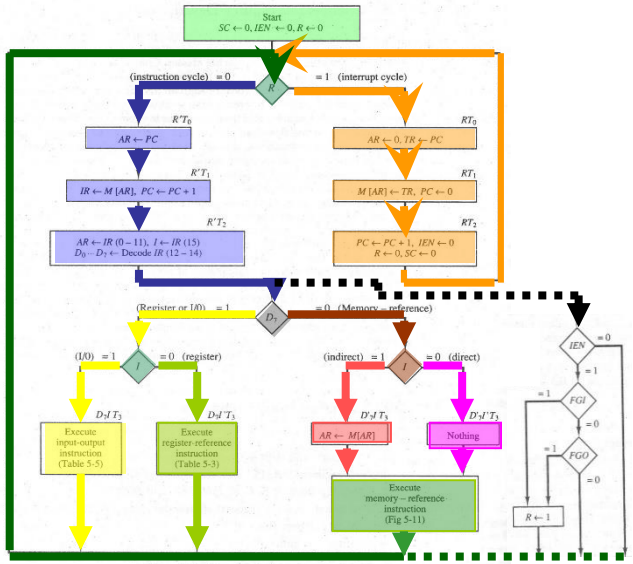


Figure 5-15 Flowchart for computer operation.

القلاب	العمل	الاستخدام
I	الخانة 15 من مسجل التعليمة IR	لتحديد نوع العنونة (مباشرة / غير مباشر) وتحديد التعليمة (مسجل /دخل خرج)
S	Start – Stop Flip Flop	يستخدم للإشارة على استمرارية تنفيذ التعليمات ويستخدم مع HLT
E	علم الحمل Carry Flag	لتحديد حدوث الحمل أولا ويستخدم في تعليمات المسجل بشكل خاص
R	قلاب في بداية المخطط	لتوضيح فيما إذا كنا سيدخل في دورة التعليمة او المقاطعة
IEN	قلاب في دورة التعليمة	لتحديد امكانية المقاطعة والسماح بها أم عدمة
FGI	قلاب في دورة التعليمة	لوصف حالة INPR ولتحديد وضع R الجديدة
FGO	قلاب في دورة التعليمة	لوصف حالة OUTR وتحديد وضع R الجديدة

يمكن دمج مجموعة تعليمات الذاكرة مع تعليمات المسجل وتعليمات الإدخال والإخراج لتصبح كالجدول (4-6) ونلاحظ أن لغة نقل المسجل المستخدمة في الجدول مفيدة ليس فقط من أجل توصيف عملية التنظيم الداخلي للحاسب الرقمية لكن أيضا تفيد في عملية تصميم الدارات المنطقية اللازمة لبناء الحاسب .

Table 4-6 Control Functions and Microoperations for the Basic Computer

Fetch	$R'T_0$:	$AR \leftarrow PC$
	$R'T_1$:	$IR \leftarrow M[AR],\quad PC \leftarrow PC + 1$
Decode	$R'T_2$:	$D_0, \ldots, D_7 \leftarrow$ Decode $IR(12\text{–}14),$
		$AR \leftarrow IR(0\text{–}11),\quad I \leftarrow IR(15)$
Indirect	$D_7'IT_3$:	$AR \leftarrow M[AR]$
Interrupt:		
$T_0'T_1'T_2'(IEN)(FGI + FGO)$:		$R \leftarrow 1$
	RT_0:	$AR \leftarrow 0,\quad TR \leftarrow PC$
	RT_1:	$M[AR] \leftarrow TR,\quad PC \leftarrow 0$
	RT_2:	$PC \leftarrow PC + 1,\quad IEN \leftarrow 0,\quad R \leftarrow 0,\quad SC \leftarrow 0$
Memory-reference:		
AND	D_0T_4:	$DR \leftarrow M[AR]$
	D_0T_5:	$AC \leftarrow AC \wedge DR,\quad SC \leftarrow 0$
ADD	D_1T_4:	$DR \leftarrow M[AR]$
	D_1T_5:	$AC \leftarrow AC + DR,\quad E \leftarrow C_{out},\quad SC \leftarrow 0$
LDA	D_2T_4:	$DR \leftarrow M[AR]$
	D_2T_5:	$AC \leftarrow DR,\quad SC \leftarrow 0$
STA	D_3T_4:	$M[AR] \leftarrow AC,\quad SC \leftarrow 0$
BUN	D_4T_4:	$PC \leftarrow AR,\quad SC \leftarrow 0$
BSA	D_5T_4:	$M[AR] \leftarrow PC,\quad AR \leftarrow AR + 1$
	D_5T_5:	$PC \leftarrow AR,\quad SC \leftarrow 0$
ISZ	D_6T_4:	$DR \leftarrow M[AR]$
	D_6T_5:	$DR \leftarrow DR + 1$
	D_6T_6:	$M[AR] \leftarrow DR,\quad$ if $(DR = 0)$ then $(PC \leftarrow PC + 1),\quad SC \leftarrow 0$

Register-reference:

$D_7 I' T_3 = r$ (common to all register-reference instructions)

$IR(i) = B_i$ $(i = 0, 1, 2, \ldots, 11)$

	r:	$SC \leftarrow 0$
CLA	rB_{11}:	$AC \leftarrow 0$
CLE	rB_{10}:	$E \leftarrow 0$
CMA	rB_9:	$AC \leftarrow \overline{AC}$
CME	rB_8:	$E \leftarrow \overline{E}$
CIR	rB_7:	$AC \leftarrow \text{shr } AC,\ AC(15) \leftarrow E,\ E \leftarrow AC(0)$
CIL	rB_6:	$AC \leftarrow \text{shl } AC,\ AC(0) \leftarrow E,\ E \leftarrow AC(15)$
INC	rB_5:	$AC \leftarrow AC + 1$
SPA	rB_4:	If $(AC(15) = 0)$ then $(PC \leftarrow PC + 1)$
SNA	rB_3:	If $(AC(15) = 1)$ then $(PC \leftarrow PC + 1)$
SZA	rB_2:	If $(AC = 0)$ then $PC \leftarrow PC + 1)$
SZE	rB_1:	If $(E = 0)$ then $(PC \leftarrow PC + 1)$
HLT	rB_0:	$S \leftarrow 0$

Input-output:

$D_7 I T_3 = p$ (common to all input–output instructions)

$IR(i) = B_i$ $(i = 6, 7, 8, 9, 10, 11)$

	p:	$SC \leftarrow 0$
INP	pB_{11}:	$AC(0\text{–}7) \leftarrow INPR,\ FGI \leftarrow 0$
OUT	pB_{10}:	$OUTR \leftarrow AC(0\text{–}7),\ FGO \leftarrow 0$
SKI	pB_9:	If $(FGI = 1)$ then $(PC \leftarrow PC + 1)$
SKO	pB_8:	If $(FGO = 1)$ then $(PC \leftarrow PC + 1)$
ION	pB_7:	$IEN \leftarrow 1$
IOF	pB_6:	$IEN \leftarrow 0$

9- تصميم الحاسب الافتراضي Design Of Basic Computer

مما مضى نستنتج أن المكونات الصلبة للحاسب الافتراضي هي :

1. وحدة ذاكرة 4096 x 16 Bit
2. تسعة مسجلات : SC , INPR , OUTR , TR , IR , AC , DR , PC , AR
3. سبعة قلابات : FGO , FGI , IEN , R , E , S , I
4. مفكك شفرة : 3 x 8 و 4 x 16 للتحكم والتزامن (D0 ÷D7) و(T0÷T15)
5. ممر مشترك بطول 16 Bit.
6. بوابات منطقية للتحكم .
7. دارة جامع ودارة منطقية موصولة إلى AC

وقد تم شرح مكونات الفقرات " I إلى "5" خلال الفصلين الأول والثاني .

R

بوابات التحكم المنطقية Control Logic Gates:

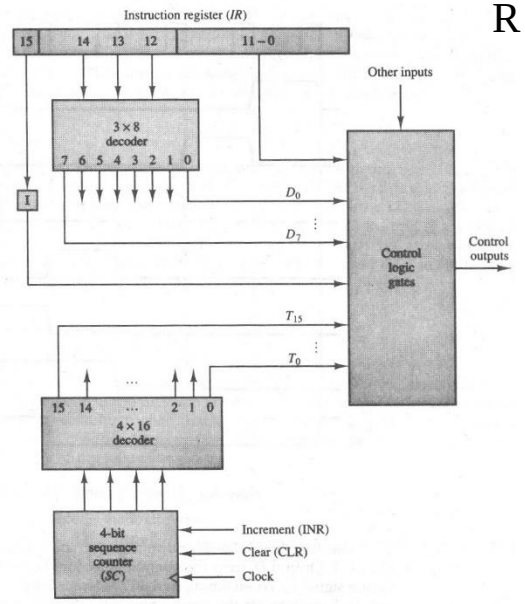


Figure 4-15 Control unit of basic computer.

وقد تم وضع المخطط الصندوقي لها من الشكل (4-6) ووظيفة هذه الدارات ما يلي:

1. توليد إشارات التحكم لمداخل المسجلات التسعة. (وهي Lood- ncrement- clear)
2. توليد إشارات التحكم للقراءة والكتابة في الذاكرة. (وهي write-Read)
3. توليد إشارات لوضع 1 و 0 أو المتمم للقلابات .
4. توليد إشارات S0, S1, S2 لاختيار المسجل على الممر المشترك أو الذاكرة
5. توليد إشارات التحكم لدارة الجامع والدارة المنطقية .

- التحكم بالمسجلات والذاكرة :

يمكن تصميم دارة التحكم لأي مسجل والتي تقوم بالتحكم بالمداخل INC ,CLR ,LD عن طريق إيجاد عبارات التحم التي تتعرض لهذا المسجل وذلك بالاستعانة بالجدول (6-4)

فمثلا إذا أردنا تصميم دارة التحكم للمسجل AR (مسجل العنوان) وبالعودة للجدول (6-4) نجد أن العبارات التي تتعرض لـ AR يمكن حصرها بالعبارات الخمسة التالية :

$$
\begin{aligned}
R'T_0: &\quad AR \leftarrow PC \\
R'T_2: &\quad AR \leftarrow IR(0\text{–}11) \\
D_7'IT_3: &\quad AR \leftarrow M[AR] \\
RT_0: &\quad AR \leftarrow 0 \\
D_5T_4: &\quad AR \leftarrow AR + 1
\end{aligned}
$$

ويمكن استنتاج معادلات التحكم منها حيث نلاحظ أن أول ثلاث عبارات تصنف على أنها تحميل Load إلى AR

أما العبارة الرابعة فهي تصفير AR (Clear)

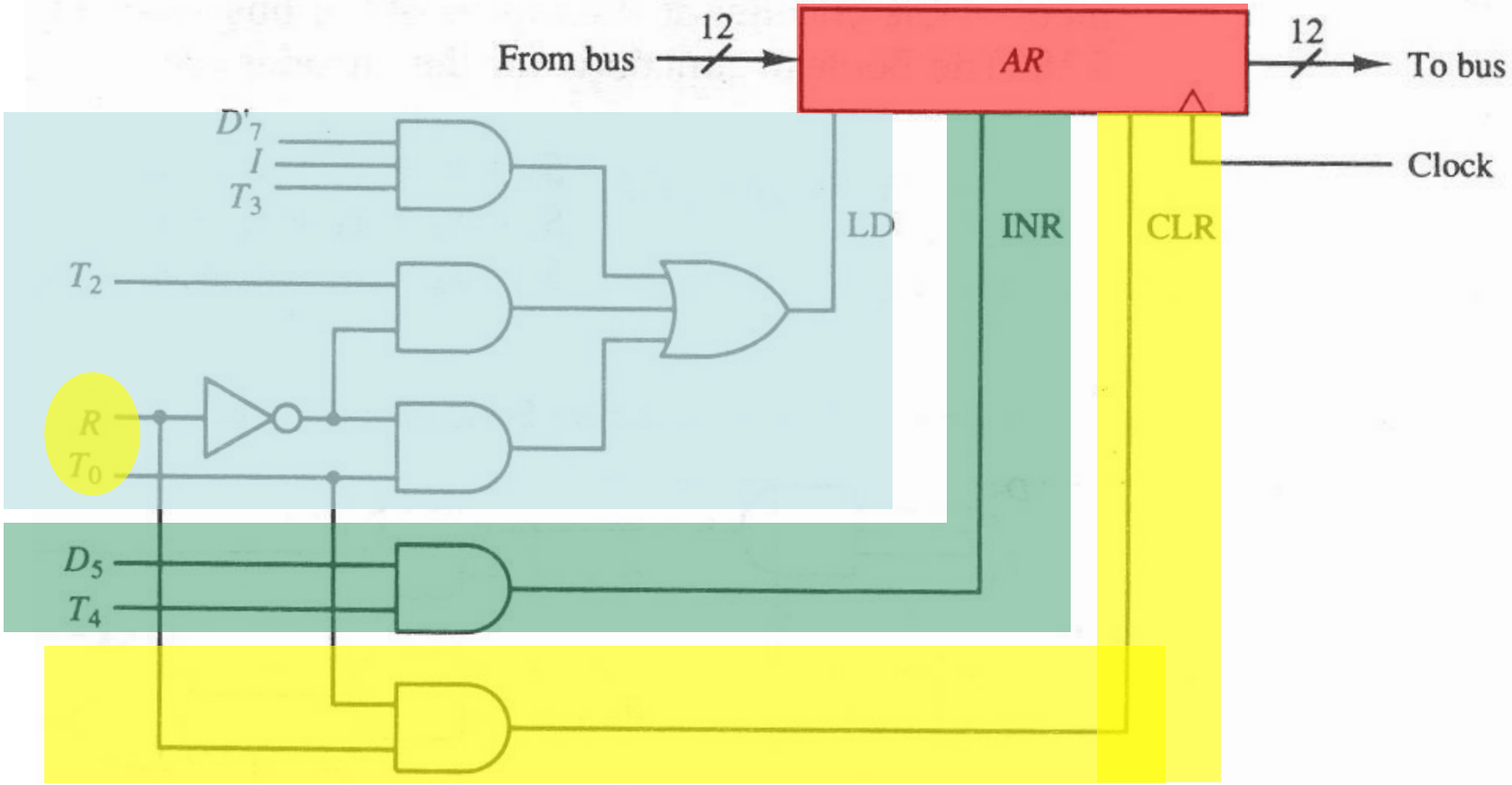
والعبارة الخامسة هي زيادة 1(Increment) وتصبح معادلات التحكم :

$$
\begin{aligned}
\text{LD}(AR) &= R'T_0 + R'T_2 + D_7'IT_3 \\
\text{CLR}(AR) &= RT_0 \\
\text{INR}(AR) &= D_5T_4
\end{aligned}
$$

$$LD(AR) = R'T_0 + R'T_2 + D'_7IT_3$$
$$CLR(AR) = RT_0$$
$$INR(AR) = D_5T_4$$

يمكن اعتماد نفس الطريقة للحصول على بوابات التحكم لمدخلي القراءة والكتابة ،
مثلا" نجد ان مدخل القراءة هو:

Figure 4-16 Control gates associated with AR.



أما الذاكرة فيمكن اعتماد نفس الطريقة للحصول على بوابات التحكم لمدخلي القراءة والكتابة فمثل نجد ان مدخل القراءة هو (من الجدول (6-4) المثل لتوصيف الكامل للحاسب الافتراضي وذلك بعد عملية جرده وحصرا جميع العمليات الخاصة بالقراءة من الذاكرة):

$$Read = R`T1 + D`7T3 + (D0 + D1 + D2 + D6)T4$$

$$Write = RT1 + (D3 + D5)\ T4 + D6\ T6$$

التحكم بقلاب وحيد Control Of Single Filp-Flop:

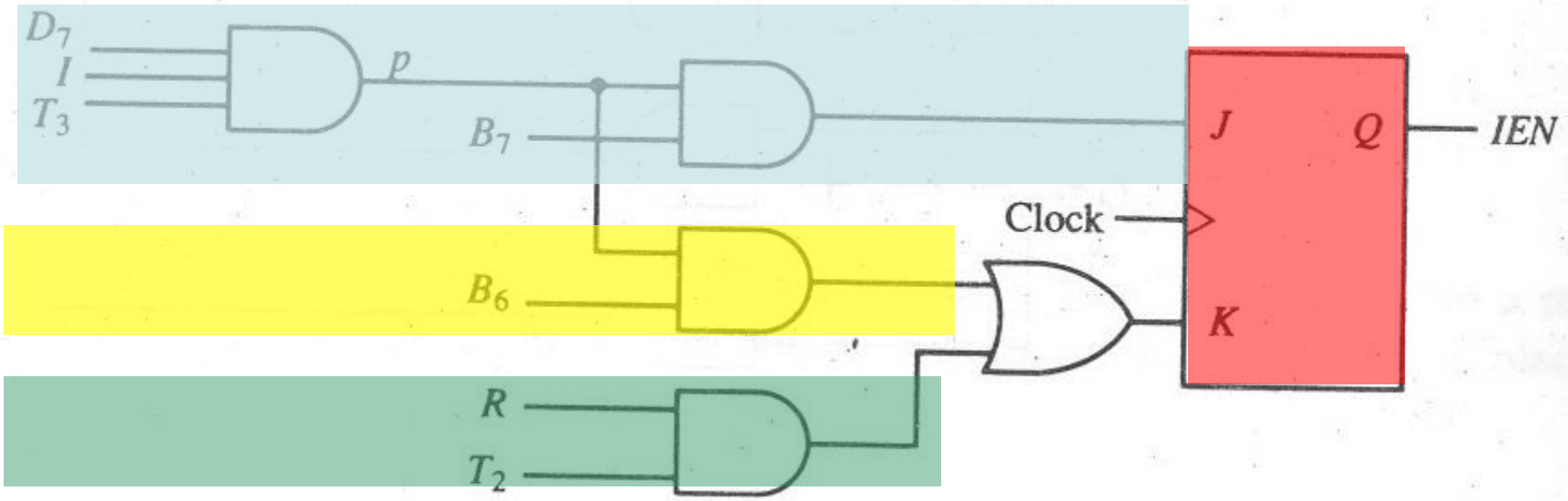
يتم تصميم بوابات التحكم للقلابات السبعة بنفس الطريقة السابقة ،

فمثلا لتصميم دارة التحكم بالقلاب IEN نعود إلى الجدول ونستخرج عبارات التحكم التي نتعامل معه ونجدها :

pB7: IEN ← 1 ; p = D7IT3
pB6: IEN ← 0
RT2: IEN ← 0

ومنه نحصل على دارة التحكم المبينة في الشكل (4-17) .

Figure 4-17 Control inputs for *IEN*.



- التحكم بالممر المشترك :

لقد وجدنا أن اختيار المسجل أو الذاكرة التي ستضع (أو يضع) قيمة على الممر المشترك يتم عبر ناخب له مداخل اختيار S2,S1,S0, ولهذه الغاية نستخدم سبعة مداخل من x1 حتى X7 مطبقة على مشفر له المخارج S0, S1, S2 ويعمل كالجدول (4-7)

Table 4-7 Encoder for Bus Selection Circuit

Inputs							Outputs			Register selected for bus
x_1	x_2	x_3	x_4	x_5	x_6	x_7	S_2	S_1	S_0	
0	0	0	0	0	0	0	0	0	0	None
1	0	0	0	0	0	0	0	0	1	*AR*
0	1	0	0	0	0	0	0	1	0	*PC*
0	0	1	0	0	0	0	0	1	1	*DR*
0	0	0	1	0	0	0	1	0	0	*AC*
0	0	0	0	1	0	0	1	0	1	*IR*
0	0	0	0	0	1	0	1	1	0	*TR*
0	0	0	0	0	0	1	1	1	1	Memory

وبالتالي يكون :

S0=X1+X3+X5+X7
S1=X2+X3+X6+X7
S2=X4+X5+X6+X7

وبالعودة إلى جدول تعليمات نقل المسجل يمكن استنتاج قيم X ،فمثلا تكون X1=1 عندما يضع AR قيمته على الممر وبالنظر إلى جدول التعليمات نجد أن ذلك يتم في الحالتين التاليتين :

$$D4T4 : PC \leftarrow AR$$
$$D5T5 : PC \leftarrow AR$$

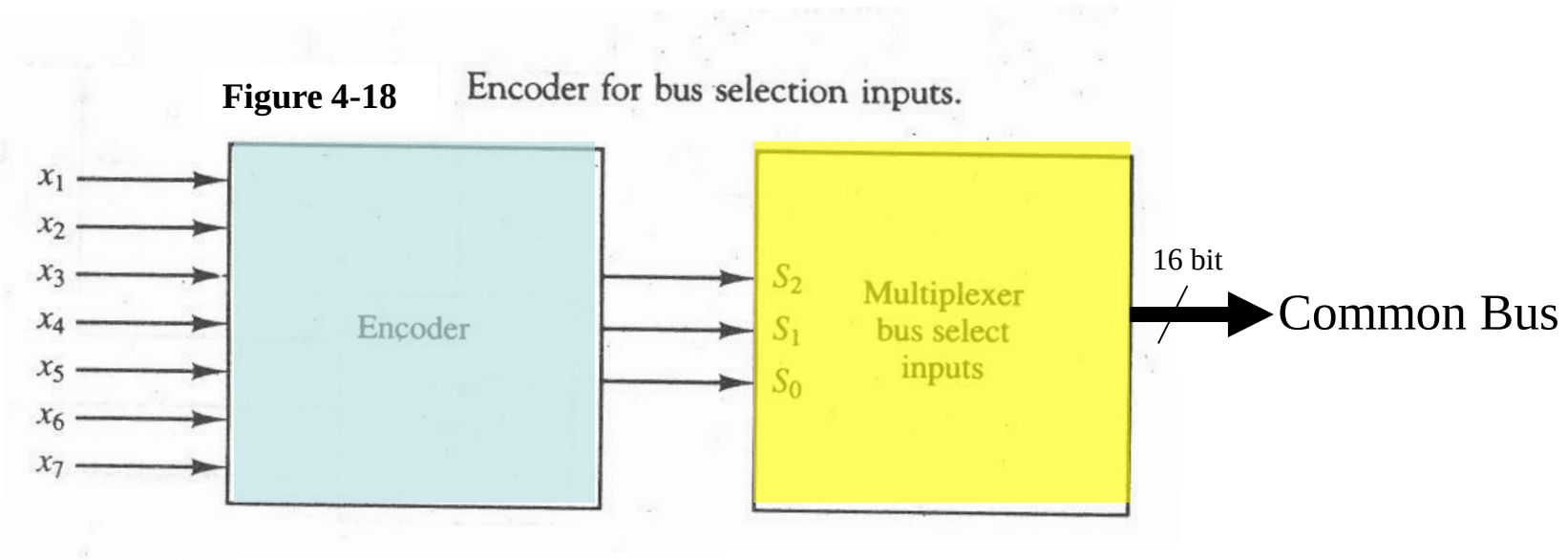
وبالتالي يكون

$$X1=D4T4 + D5T5$$

مثال :

$$X7=R`T1+D`7\ I\ T3 + (D0 + D1 + D2 + D6 +)T4$$

يكون X7 = 1 عندما الذاكرة محتوياتها على الممر المشترك (قراءة)



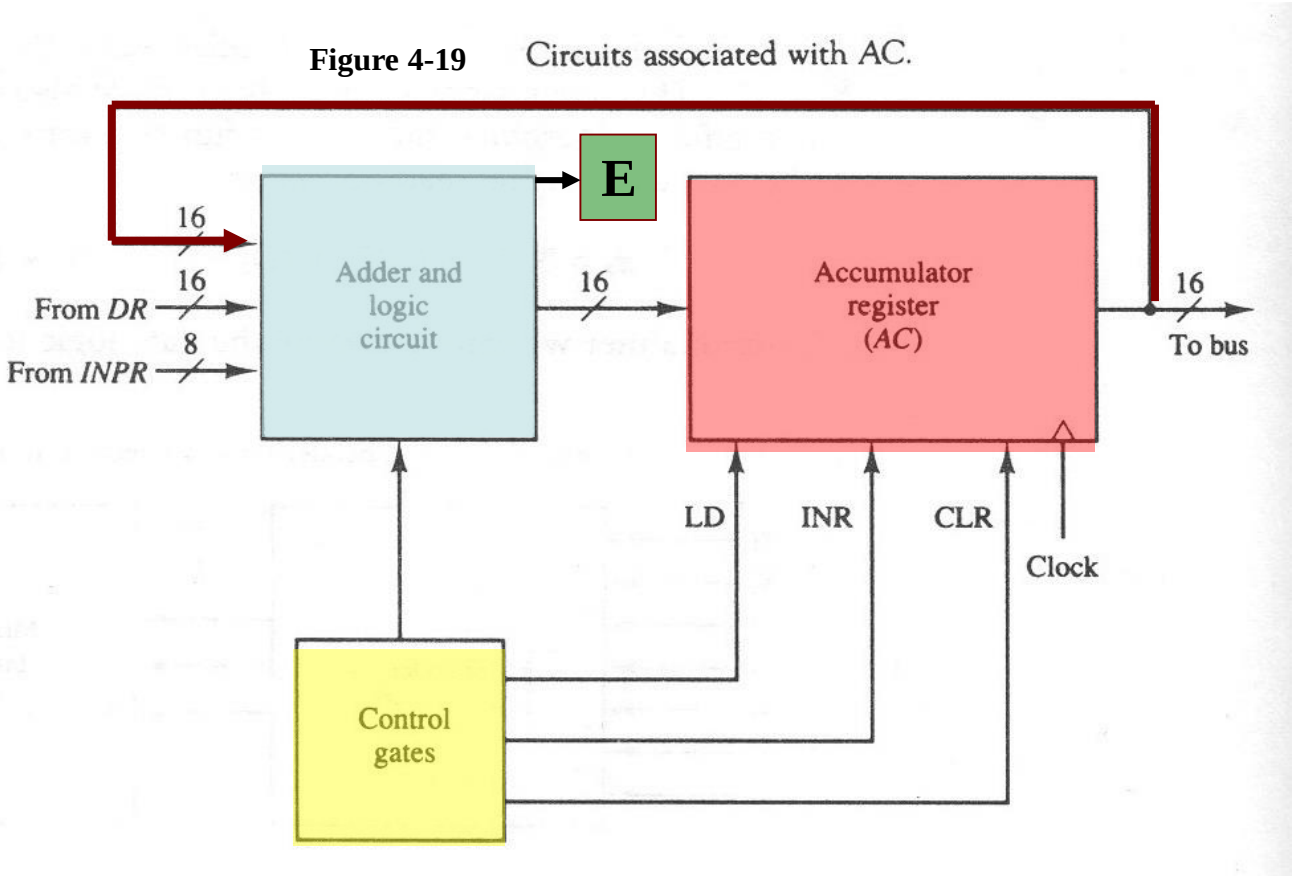
10- تصميم منطق التحكم :

يبين الشكل (4-19) المخطط الصندوقي لدارة التحكم المشاركة للمراكم والتي تتألف من المراكم نفسه بالإضافة إلى دارة جامع ومنطق

تتألف من ثلاثة مداخل الأول من المراكم والثاني من DR والثالث من INPR ولها مخرج إلى المراكم، كذالك توجد بوابات تحكم تقوم بتوليد إشارات التحكم بالمراكم (إشارات CLR , INR , LD)

وبدارة الجامع والمنطق .

Figure 4-19 Circuits associated with AC.



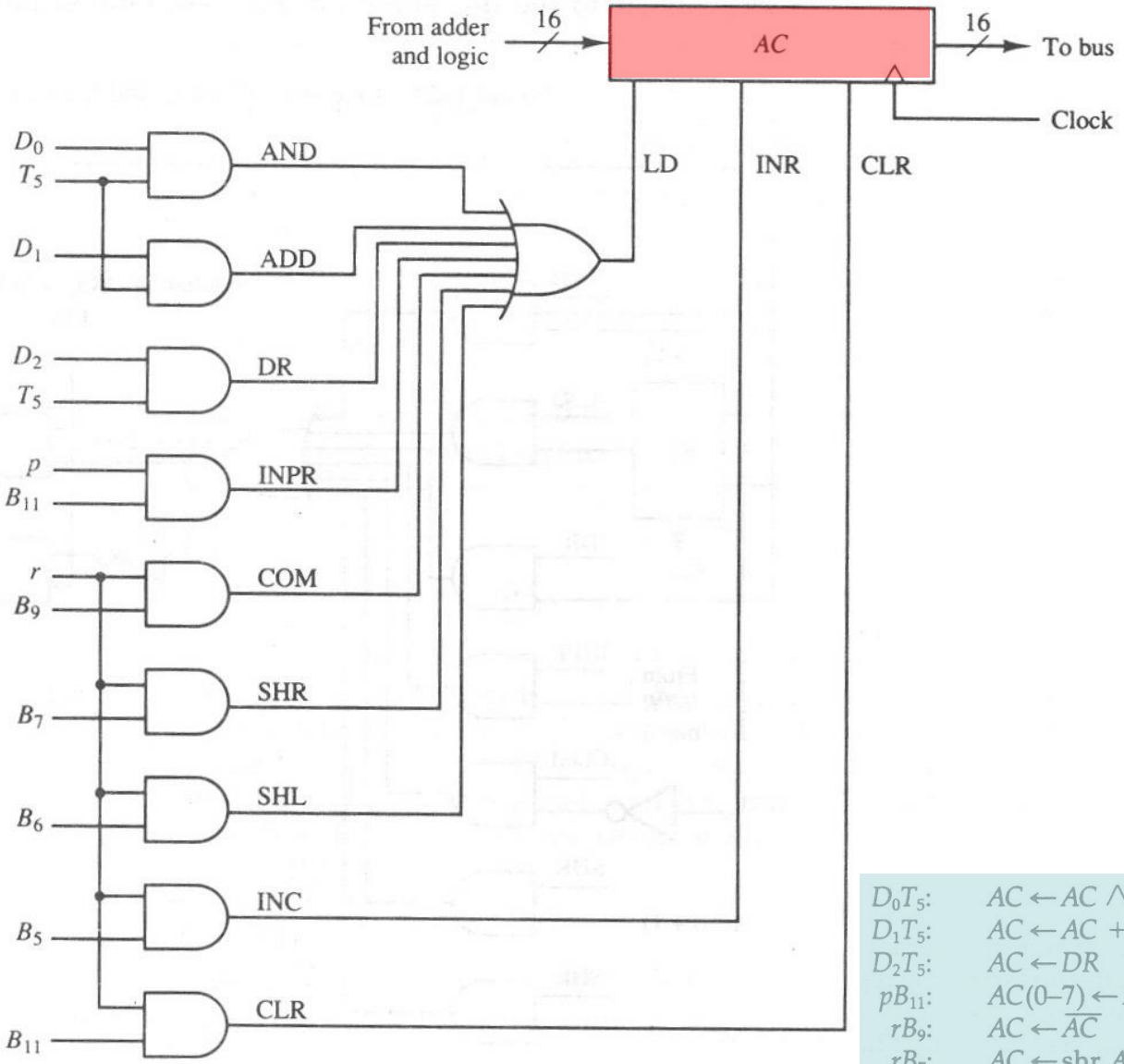
نستخرج تعليمات نقل المسجل التي نتعامل مع المراكم فنجدها التالية :

D_0T_5:	$AC \leftarrow AC \wedge DR$	AND with *DR*
D_1T_5:	$AC \leftarrow AC + DR$	Add with *DR*
D_2T_5:	$AC \leftarrow DR$	Transfer from *DR*
pB_{11}:	$AC(0\text{–}7) \leftarrow INPR$	Transfer from *INPR*
rB_9:	$AC \leftarrow \overline{AC}$	Complement
rB_7:	$AC \leftarrow \text{shr } AC,\ AC(15) \leftarrow E$	Shift right
rB_6:	$AC \leftarrow \text{shl } AC,\ AC(0) \leftarrow E$	Shift left
rB_{11}:	$AC \leftarrow 0$	Clear
rB_5:	$AC \leftarrow AC + 1$	Increment

بالتالي يمكن تصميم دارة بوابات التحكم بالمراكم لتصبح كما الشكل (4-20) حيث تطبق على مدخل LD إشارة فعال "1" عند :

- تنفيذ تعليمة AND وبتحقق الشرط D0T5
- أو عند التعليمة ADD وبتحقق D1T5
- أو عند نقل DR وبتحقق D2T5 وعند نقل INPR وبتحقق PB11
- أو عند إتمام المراكم وبتحقق Rb9 أو عند إزاحة المراكم لليمين وبتحقق RB7 أو لليسار بتحقق RB6

Figure 4-20 Gate structure for controlling the LD, INR, and CLR of AC.



بينما يطبق 1" على مدخل CLR عند تنفيذ تعليمة تصفير المراكم وبتحقق RB11 بينما يطبق "1 " على مدخل INR عند تنفيذ زيارة واحد للمراكم وبتحقق RB6 .

R

D_0T_5:	$AC \leftarrow AC \wedge DR$	AND with DR
D_1T_5:	$AC \leftarrow AC + DR$	Add with DR
D_2T_5:	$AC \leftarrow DR$	Transfer from DR
pB_{11}:	$AC(0\text{–}7) \leftarrow INPR$	Transfer from $INPR$
rB_9:	$AC \leftarrow \overline{AC}$	Complement
rB_7:	$AC \leftarrow \text{shr } AC,\ AC(15) \leftarrow E$	Shift right
rB_6:	$AC \leftarrow \text{shl } AC,\ AC(0) \leftarrow E$	Shift left
rB_{11}:	$AC \leftarrow 0$	Clear
rB_5:	$AC \leftarrow AC + 1$	Increment

يتم تصميم دارة الجامع والمنطق بنفس الطريقة وباستخدام خطوط مخارج بوابات AND للشكل (4-20) والتي تدخل تحت التسميات (AND , ADD , DR , INPR , COM , SHR , SHL)

يبين الشكل (4-21) تصميم دارة جامع ومنطق لخانة واحدة i.

Figure 4-21 One stage of adder and logic circuit.