تصميم وحدة التحكم
Control Unit (2)

م. عبير ميّا م. مصعب خباز

2/11/2022

تكملة حل السؤال من المحاضرة السابقة

اكتب مراحل تنفيذ التعليمات التسعة السابقة بلغة نقل السجلات $RTL\ (Register\ Transfer\ Language)$

والآن بعد أن تعرفنا على مرحلة جلب التعليمة $fetch$ والتي تمر بثلاث مراحل

$$F_1: AR \leftarrow PC$$
$$F_2: DR \leftarrow M[AR]; PC \leftarrow PC + 1$$
$$F_3: IR \leftarrow DR(op); AR \leftarrow DR(x)$$

1. $MOV\_R \Leftrightarrow R \leftarrow AC$
2. $SUB\_R \Leftrightarrow AC \leftarrow AC - R$
3. $OR\_R \Leftrightarrow AC \leftarrow AC\ or\ R$
4. $LDR\_D\ x \Leftrightarrow R \leftarrow M[x]$
5. $STAC\_D\ x \Leftrightarrow M[x] \leftarrow AC$
6. $STR\_ID\ x \Leftrightarrow M[M[x]] \leftarrow R$
7. $SUB\_D\ x \Leftrightarrow AC \leftarrow AC - M[x]$
8. $JMP\_D\ x \Leftrightarrow Jump\ to\ direct\ address$
9. $NOP \Leftrightarrow No\ Operation\ (Do\ Nothing)$

- وعلمنا بأن جميع التعليمات تمر بهذه المراحل الثلاث لجلبها من الذاكرة.
- الآن سنقوم بمعرفة عدد مراحل كل تعليمة من التعليمات التسعة السابقة ونقصد هنا بعدد مراحل التعليمة أيّ عدد تعليمات $fetch$ بالإضافة إلى عدد مراحل تنفيذ هذه التعليمة.

عدد مراحل التعليمة $= 3\ (fetch)$ + عدد مراحل التنفيذ

تذكر

- أي عنوان ذاكرة يجب التعامل معه عن طريق AR.
- أي قيمة نقوم بجلبها من الذاكرة أوكتابتها بالذاكرة $(data)$ يجب أن تمر على DR.

1. $MOV\_R: R \leftarrow AC$

في هذه التعليمة سنقوم فقط بنقل محتوى المراكم AC إلى السجل R أيّ ستنفذ في مرحلة واحدة بالإضافة إلى 3 مراحل الـ$fetch$ (فتكون عدد مراحل التعليمة $MOV\_R = 4$ مراحل)

2. $SUB\_R: AC \leftarrow AC - R$

بالبداية سيكون عدد مراحل هذه التعليمة 3 وهي عدد مراحل الـ$fetch$ ولكن نلاحظ بأن هذه المراحل غير كافية لأن التعليمة لم تتنفذ بعد فمرحلة التنفيذ تتلخص بطرح محتوى السجل R من المراكم AC ووضعها في الـAC (فتكون عدد مراحل تعليمة $SUB\_R = 4$ مراحل).

3. $OR\_R: AC \leftarrow AC \ or \ R$

كما فعلنا في التعليمة السابقة سنجد أن مراحل هذه التعليمة هي 4 أيضاً.

- ملاحظة: في التعليمات السابقة لم نستخدم AR أو DR لأن التعديلات ستكون على السجلات فقط لم نضطر للوصول للذاكرة بالإضافة إلى أن مرحلة تنفيذ التعليمة كانت على مرحلة واحدة.

4. $LDR\_D\ x: R \leftarrow M[x]$

بالإضافة إلى مراحل $fetch$ ستكون مراحل تنفيذ التعليمة على الشكل:

$$LDR\_D_1: DR \leftarrow M[AR]$$
$$LDR\_D_2: R \leftarrow DR$$

هنا قمنا باستبدال x (عنوان الذاكرة) بـAR لماذا؟؟ (لأن أيّ عنوان ذاكرة نتعامل معه عن طريق AR) ومن ثم وضعنا محتوى عنوان الذاكرة (الـdata) فيDR (أيّ قيمة نقوم بجلبها أو وضعها في الذاكرة يكون عن طريق الـDR) ومن ثم نضع محتوى الـDR في السجل R لإكمال تنفيذ التعليمة.

الآن نجد بأنه لدينا مرحلتين تنفيذ بالإضافة إلى مراحل $fetch$ ⇐ عدد مراحل تعليمة $LDR\_D = 5$ مراحل.

5. $STAC\_D\ x: M[x] \leftarrow AC$

مراحل التنفيذ على الشكل:

بداية قمنا بنقل محتوى الـAC للـDR لأنها معطيات ونريد وضعها في عنوان في الذاكرة ومن ثم قمنا بنقل محتوى الـDR لعنوان الذاكرة.

$$STAC\_D_1: DR \leftarrow AC$$
$$STAC\_D_2: M[AR] \leftarrow DR$$

(وبالتالي تكون عدد مراحل التعليمة $STAC\_D$ هي 5 مراحل)

6. $STR\_ID\ x: M[M[x]] \leftarrow R$

هنا نجد بأن العنونة غير مباشرة هذا يعني بأن محتوى $M[M[x]]$ أيّ $M[x]$ هو أيضاً عنوان ذاكرة بداخله المحتوى الذي نريد استبداله بمحتوى السجل R .

$$STR\_ID_1: DR \leftarrow M[AR]$$
$$STR\_ID_2: AR \leftarrow DR$$
$$STR\_ID_3: DR \leftarrow R$$
$$STR\_ID_4: M[AR] \leftarrow DR$$

- شرح الخطوات: بداية قمنا بوضع محتوى عنوان الذاكرة في الـDRومن ثم قمنا بوضع محتوى الـDR بالـAR (وذلك لأن DR حالياً يحتوي على عنوان ذاكرة) ثم نقوم بوضع محتوى السجل R بالـDR لأنها معطيات ونريد وضعها في عنوان ذاكرة ثم نقوم بوضع الـDR في عنوان الذاكرة المطلوب وبالتالي عدد مراحل التعليمة هي 7.

7. $SUB\_D\ x: AC \leftarrow AC - M[x]$

تكون مراحل التنفيذ بالشكل:

$$SUB\_D_1: DR \leftarrow M[AR]$$
$$SUB\_D_2: AC \leftarrow AC - DR$$

نقوم بوضع محتوى عنوان الذاكرة $M[AR]$ في الـDR (قيمة قمنا بجلبها من الذاكرة) ومن ثم قمنا بطرح محتوى الـDR من المراكم AC ومن ثم وضع الناتج في AC .

فتكون عدد مراحل هذه التعليمة هي 5 مراحل.

8. $JMP\_D\ x: Jump\ to\ direct\ address$

هنا العنونة مباشرة أيّ $M[x]$ فتكون مراحل التنفيذ بالشكل:

$$JMP\_D_1: DR \leftarrow M[AR]$$
$$JMP\_D_2: PC \leftarrow DR(x)$$

قمنا بوضع محتوى عنوان الذاكرة $M[AR]$ في الـDR ومن ثم نقلنا محتوى الـDR للـPC فتكون عدد المراحل تساوي 5 مراحل.

تختلف تعليمة JMP بحسب نوع العنونة في التعليمة السابقة كانت العنونة مباشرة أما في حال كانت العنونة فورية فسيكون عدد المراحل يساوي 4 مراحل.

$$JMP\ 64: PC \leftarrow 64$$

ستكون مراحل التعليمة هي تعليمة التنفيذ (وضع الثابت 64 في الـPC) بالإضافة إلى تعليمات $fetch \Leftarrow 4$ مراحل

$$JMP: PC \leftarrow DR(x)$$

أما في حال كانت العنونة غير مباشرة $JMP\ ID\ x$ أيّ $M[M[x]]$

$$JMP\_ID_1: DR \leftarrow M[AR]$$
$$JMP\_ID_2: AR \leftarrow DR$$
$$JMP\_ID_3: DR \leftarrow M[AR]$$
$$JMP\_ID_4: PC \leftarrow DR$$

- وبالتالي سيكون عدد المراحل هو 7 مراحل.

9. $NOP: No\ Operation\ (Do\ Nothing)$

- أيضاً مراحل هذه التعليمة ستكون 4 وذلك لأنه أول 3 مراحل هي $fetch$ لجلب التعليمة من الذاكرة والمراحل التالية ستكون لتحليل التعليمة وتنفيذها وحفظ النواتج وبالتالي احتجنا هنا لمرحلة رابعة لتحليل التعليمة $Decode$ حتى لو لم تكن تقوم بأيّ شيء كتنفيذ.

3. بفرض كانت القيم الابتدائية للسجل $PC = 0x01B$ وكان محتوى الكلمة الذاكرية ذات العنوان $0x01B$ هو ترميز التعليمة $LDR\_D\ x$ ما هي قيم السجلات PC, AR, DR, IR خلال كل مرحلة من مراحل جلب وتنفيذ التعليمة $LDR\_D\ 0x01E$؟

$Address$	018	019	$01A$	$01B$	$01C$	$01D$	$01E$	$01F$
$Content$	$901E$	$001F$	$01B5$	$601E$	$1D05$	0020	3020	2011

$state$	PC	AR	IR	DR	R
$F_1: AR \leftarrow PC$ نسخ محتوى الـPC للسجل AR	$0x01B$	$0x01B$	–	–	–
$F_2: DR \leftarrow M[AR];$ نقل محتوى العنوان AR للسجل DR $PC \leftarrow PC + 1$ زيادة محتوى العداد بمقدار 1	$0x01B + 1 = 0x01C$	$0x01B$	–	$0x601E$	–
$F_3: IR \leftarrow DR(op);$ $AR \leftarrow DR(x)$	$0x01C$	$0x01E$	$0x6$	$0x601E$	–
$LDR\ D_1: DR \leftarrow M[AR]$ وضع محتوى العنوان AR الموجود بالذاكرة(من جدول الذاكرة) بالسجل DR.	$0x01C$	$0x01E$	$0x6$	$0x3020$	–
$LDR\ D_2: R \leftarrow DR$ نسخ محتوى السجل DR ووضعه في السجل R.	$0x01C$	$0x01E$	$0x6$	$0x3020$	$0x3020$

- ملاحظات

$$\underbrace{0x}_{DR(op)}\underbrace{601E}_{DR(x)}$$

$0x6$ $0x01E$

- عندما نجد $M[AR]$ لا ننظر لمحتوى السجل AR في الجدول نقوم بالنظر في جدول الذاكرة على محتوى العنوان AR.

مثال:

في الذاكرة	في الجدول
$01E: 3020$	$AR: 0x01E$

ولدينا التعليمة $DR \leftarrow M[AR]$

فتصبح الـDR $DR: 0x3020$

وغالبا ما نتعامل مع الـAR فقط كعنوان ذاكرة وبداخله محتوى أيّ بالشكل $M[AR]$ ونتعامل معه كسجل كما في التعليمة $AR \leftarrow PC$ أو $AR \leftarrow DR$.

- لا ننسى بأننا نتعامل بالنظام السداسي عشر بالتالي علينا أن نراعي ذلك في العمليات كالجمع والطرح وغيرها.
- في الجدول السطور الثلاثة الأولى نكتبها دائماً وهي عبارة عن تعليمات الـ$fetch$ والتي تمر بها كل تعليمة أما بعدها فتختلف التعليمة حسب المطلوب بالسؤال.

4. ارسم مخطط الحالات مع ترميز الحالات.

بداية عدد الحالات = عدد التعليمات $fetch$ (أيّ 3) + عدد مراحل التنفيذ فقط لكل تعليمة(16)

$2^4 = 16 \Rightarrow 2^5 = 32$ (لا تكفي عدد الحالات ☹)

عدد الحالات = 19 حالة ⇐ الترميز على 5 bit

معادلات الـ$fetch$

1) $F_1: AR \leftarrow PC$
2) $F_2: DR \leftarrow M[AR]; PC \leftarrow PC + 1$
3) $F_3: IR \leftarrow DR(op); AR \leftarrow DR(x)$

عدد مراحل التنفيذ لكل عملية

1) MOV_R	$R \leftarrow AC$	9) STR_ID_2	$AR \leftarrow DR$
2) SUB_R	$AC \leftarrow AC-R$	10) STR_ID_3	$DR \leftarrow R$
3) OR_R	$AC \leftarrow AC\ or\ R$	11) STR_ID_4	$M[AR] \leftarrow DR$
4) LDR_D_1	$DR \leftarrow M[AR]$	12) SUB_D_1	$DR \leftarrow M[AR]$
5) LDR_D_2	$R \leftarrow DR$	13) SUB_D_2	$AC \leftarrow AC - DR$
6) $STAC\_D_1$	$DR \leftarrow AC$	14) JMP_D_1	$DR \leftarrow M[AR]$
7) $STAC\_D_2$	$M[AR] \leftarrow DR$	15) JMP_D_2	$PC \leftarrow DR(x)$
8) STR_ID_1	$DR \leftarrow M[AR]$	16) NOP	$Do\ Nothing$

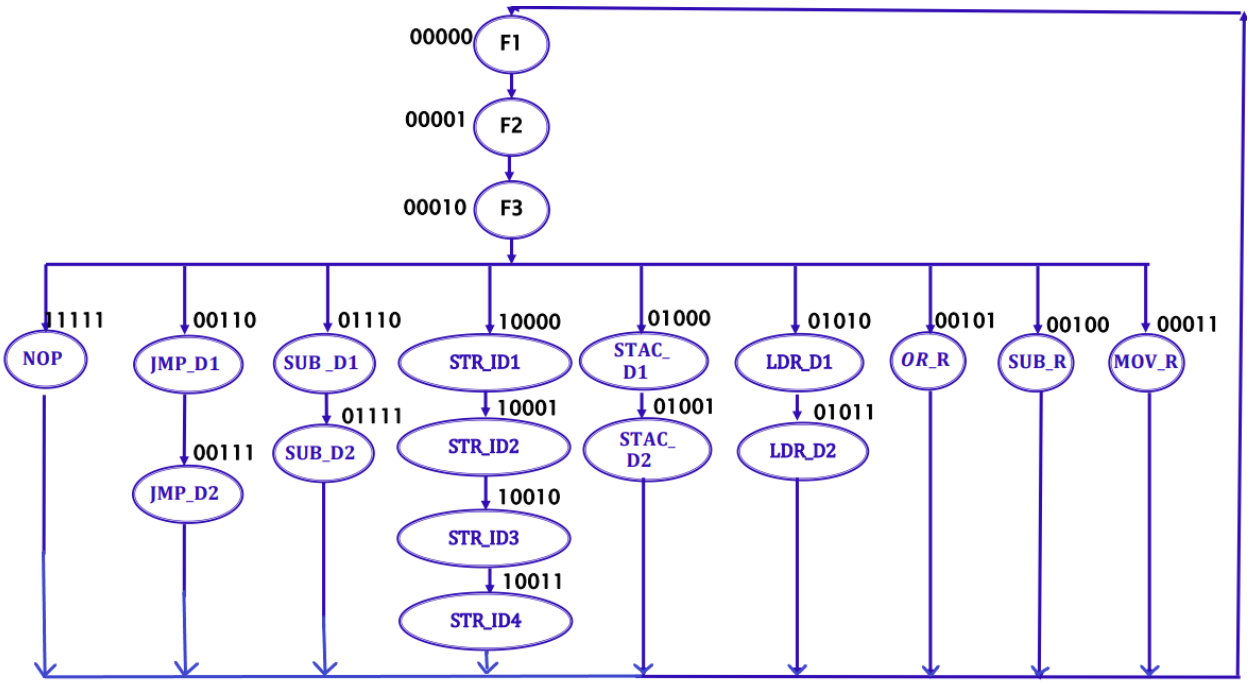
ولرسم مخطط الحالات سنقوم باستخدام عدّاداً عوضاً عن القلاب من أجل كل حالة.

العمليات التي سنطبقها على العدّاد

1. الزيادة بمقدار 1 (increment) نستخدمها عند الانتقال من حالة إلى أخرى بحيث تكون الحالات متسلسلة مثل الانتقال من $F_2 \leftarrow F_1$ حيث عمليات الـ$fetch$ متسلسلة وقريبة لا يمكن أن تسبق حالة حالة أخرى, أو الانتقال من حالة إلى أخرى ضمن نفس التعليمة.

   مثال: $STR\_ID_4 \leftarrow STR\_ID_3,\ SUB\_D_2 \leftarrow SUB\_D_1$

2. تحميل قيمة معينة على العداد $(load)$ تغيّر قيمة العداد إلى قيمة x وتستخدم بعد مرحلة F_3 للذهاب للتعليمة المناسبة إذ لا يمكن إضافة 1 لقيمة ترميز F_3 لوجود 9 حالات مختلفة.
3. تصفير العداد $(Reset)$ يستخدم بعد الانتهاء من مراحل التعليمة فيجب علينا العودة إلى F_1 لجلب التعليمة التالية.



فيكون المخطط النهائي لرسم الحالات مع ترميزها حيث عدد الحالات = 19 حالة والترميز على $5\ bit$.

- ملاحظة: ترميز العمليات فيما بعد F_3 (الـ9 حالات) اختياري بحيث يمكننا الترميز بأي طريقة منطقية بحيث يكون الترميز مختلف عن حالات $fetch$ الثلاث وأن تكون مراحل (حالات) التعليمة الواحدة متسلسلة بزيادة 1 في التعليمات التي فيها أكثر من حالة.

5. ارسم ممر المعطيات $Data\ Path$ وحدد إشارات التحكم المطلوبة ثم استنتج معادلاتها؟

في هذا الطلب سوف نحتاج إلى ثلاثة أجزاء رئيسية قبل البدء بالرسم:

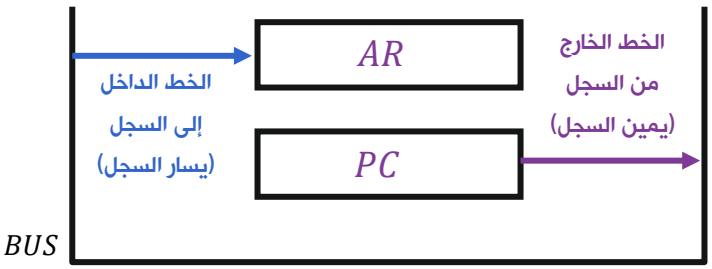
1. السجلات الموجودة في المعالج $(AR, DR, IR \cdots)$.
2. الذاكرة $(memory)$.
3. الـBus وهو طريق النقل من وإلى السجلات.

سنقوم بافتراض أن أيّ عملية نقل يجب أن تحدث باستخدام الـBus إذ لا يمكن أن نصل سجلين ببعضهما مباشرةً دون المرور بـBus.

سنقوم بالمرور على جميع الحالات الـ19 ورسم خطوط الوصل من وإلى السجلات.

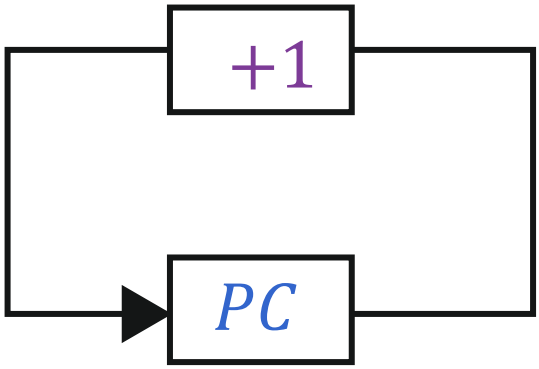
مثال كيفية الوصل باستخدام الـBUS لـ$F_1: AR \leftarrow PC$

سنفترض أن الخط الذي على يمين السجل هو الخارج من السجل والخط الذي على يسار السجل هو الداخل إلى السجل فيكون الشكل:

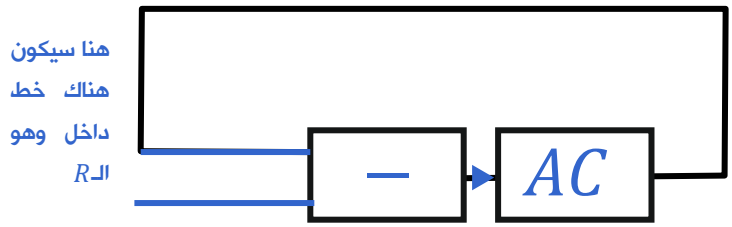


هذا في الحالات العامة ولكن لدينا بعض الحالات الخاصة

- في بعض الحالات لا داعي لخروج التعليمة إلى الـBUS إذ يكون التعديل على نفس السجل.



مثال: $PC \leftarrow PC+1$



مثال: $AC \leftarrow AC-R$

أي سنقوم برسم دارة صغيرة في نفس السجل.

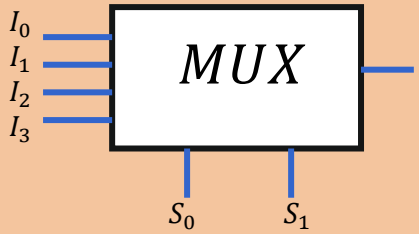
- عند وجود أكثر من دخل إلى نفس السجل يجب علينا استخدام الناخب (Multiplexer) لانتخاب خرج واحد من جميع حالات الدخل.

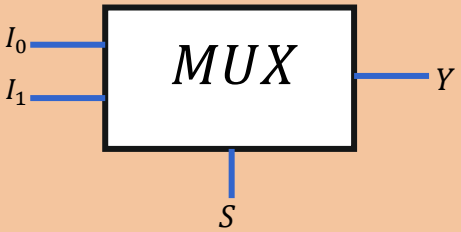
تذكر: الـMultiplexer

عدد المتحكمات: n

حالات الدخل: 2^n

حالات الخرج: 1





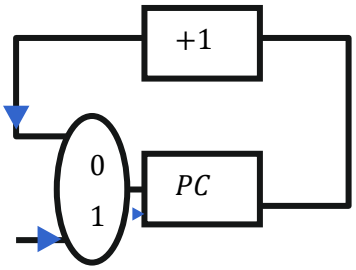
أيّ في لحظة ما يتم انتخاب حالة واحدة من حالات الدخل حسب التعليمة المراد تنفيذها وهذا ما نريده في مثالنا.

- الآن ما هي السجلات التي تحتاج إلى ناخب؟

1) PC

$PC \leftarrow PC + 1$

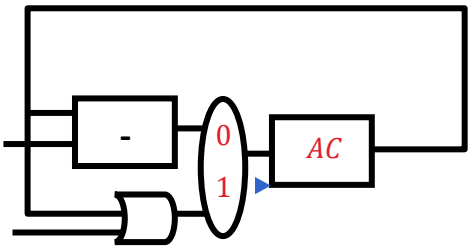
$PC \leftarrow DR(x)$



2) AC

$AC \leftarrow AC - R$

$AC \leftarrow A\ or\ R$

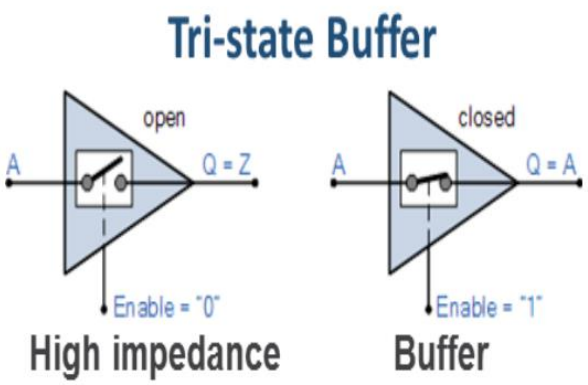


وبعد الانتهاء من رسم جميع خطوط الدخل والخرج يتبقى لدينا مشكلة واحدة وهي أن جميع السجلات موصولة مع الـBUS في نفس الوقت لذلك علينا إيجاد آلية لتنظيم دخول وخروج المعلومات من وإلى الـBUS.

مشكلة تنظيم خرج السجلات إلى الـ BUS

سنقوم باستخدام آلية مشابهة للناخب.

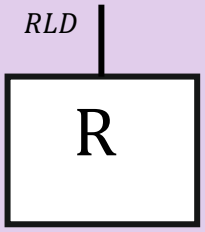
وهي الـ $Tri-State\ Buffer$ وهي عبارة عن قاطع on/off فعند لحظة ما نجعل قيمة القاطع $1\ (on)$ عند السجل الذي نتعامل معه في هذه التعليمة وباقي القواطع جميعها $0\ (off)$ وعندها نضمن تنظيم خروج المعلومات إلى الـ BUS دون حصول تداخل ونقوم بوضع الـ $Tri-State\ Buffer$ عند خرج كل سجل يتصل مع الـ BUS (R, DR, PC, AC) واثنين لتنظيم الدخول من وإلى الذاكرة.



مشكلة تنظيم دخل السجلات من الـ BUS

هنا سنقوم باستخدام خط تحكم من أجل كل سجل $(load)$ وتكون وظيفته هي السماح بالكتابة على السجل أو منع الكتابة عليه (رفض $\leftarrow 0$, سماح $\leftarrow 1$) نقوم بإرسال 1 للسجل المراد التغيير عليه حسب التعليمة و 0 للباقي.

ويكون شكله:

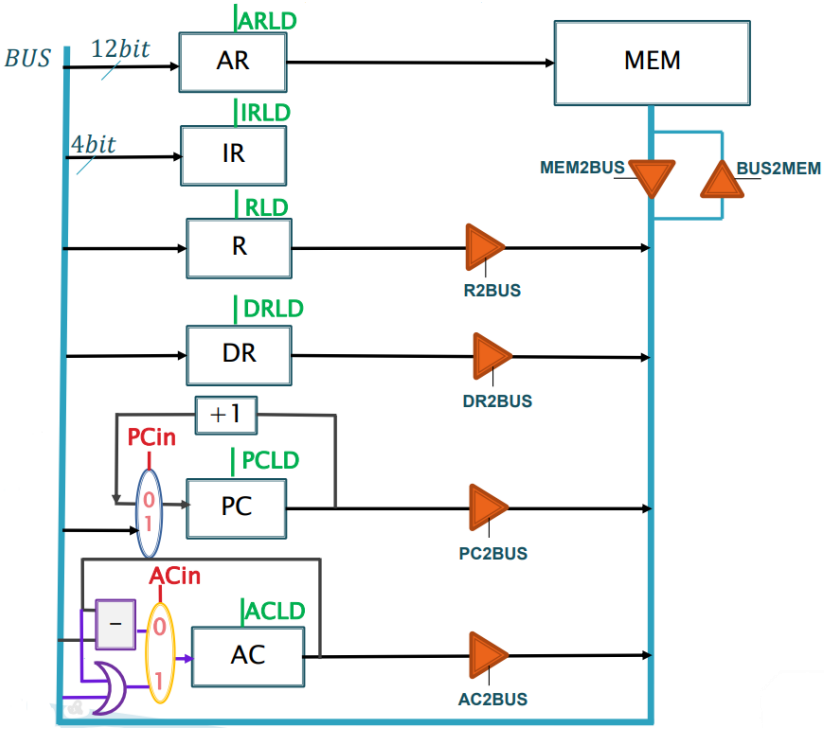


ولكن كيف نسميه؟ نضع اسم السجل وبجانبه $(load)LD$

$R \rightarrow RLD \quad DR \rightarrow DRLD \quad IR \rightarrow IRLD$.

ومنه تكون الرسمة النهائية لممر المعطيات من الشكل:

$F_1: AR \leftarrow PC$
$F_2: DR \leftarrow M[AR]; PC \leftarrow PC + 1$
$F_3: IR \leftarrow DR(OP); AR \leftarrow DR(X)$
$MOV\ R: R \leftarrow AC$
$SUB\ R: AC \leftarrow AC - R$
$OR\ R: AC \leftarrow AC\ or\ R$
$LDR\ D_1: DR \leftarrow M[AR]$
$LDR\ D_2: R \leftarrow DR$
$STAC\ D_1: DR \leftarrow AC$
$STAC\ D_2: M[AR] \leftarrow DR$
$STR\ ID1: DR \leftarrow M[AR]$
$STR\ ID_2: AR \leftarrow DR$
$STR\ ID_3: DR \leftarrow R$
$STR\ ID_4: M[AR] \leftarrow DR$
$SUB\ D_1: DR \leftarrow M[AR]$
$SUB\ D_2: AC \leftarrow AC - DR$
$JMP\ D_1: DR \leftarrow M[AR]$
$JMP\ D_2: PC \leftarrow DR(x)$
$DO\ nothing\ NOP:$



النهاية