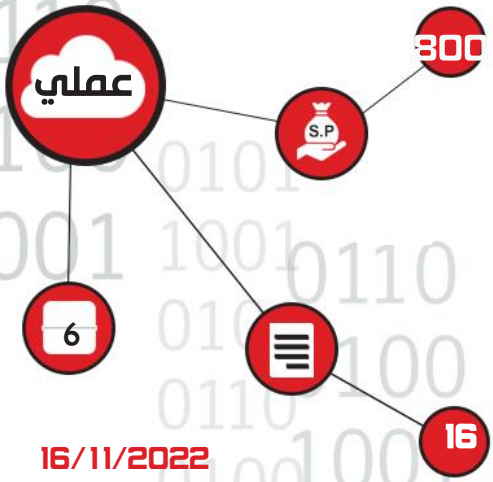


التوارد Pipelining

م. عبير ميّا م. مصعب خباز

محتوى مجاني غير مخصص للبيع التجاري

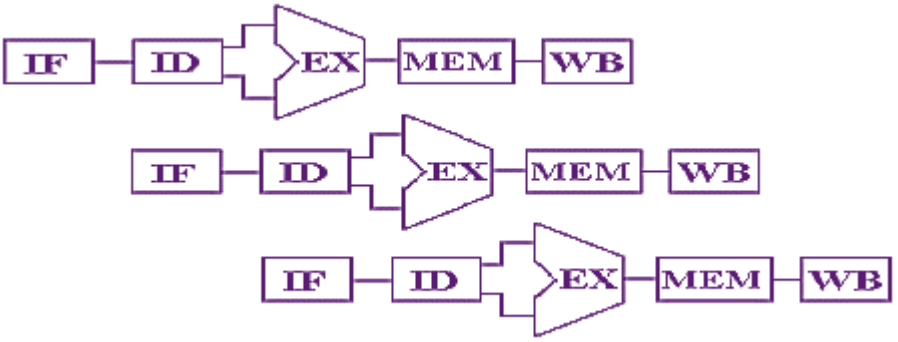
16/11/2022

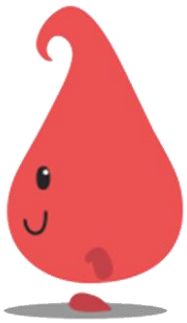
RB Informatics;

بنيان الحاسوب 2

تكلمنا في المحاضرات السابقة عن آلية تصميم ورسم وحدة التحكم Control Unit بالطريقة العتادية عن طريق معادلات إشارات التحكم. في هذه المحاضرة سنتحدث عن آلية عمل التعليمات بشكل متوارد في معالجات ال MIPS وعلى أعطال المعطيات Data Hazards وكيفية معالجتها برمجيا.

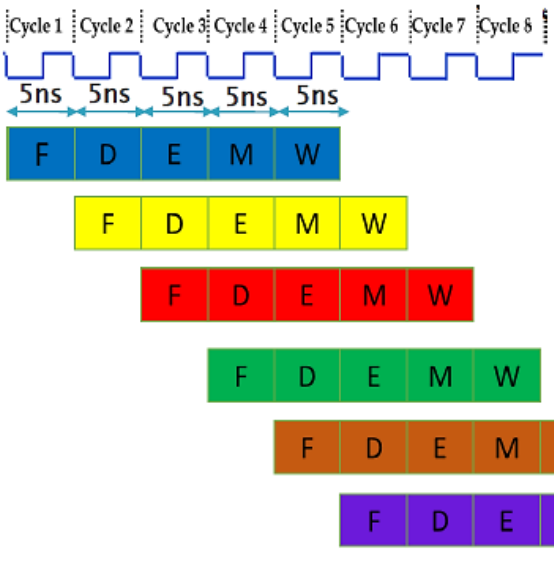
التوارد Pipelining





- **تمرين 1:** بفرض معالج متوارد $P.L$ مؤلف من 5 مراحل F, D, E, M, W وبفرض أن أزمنة التأخير كالتالي: $Regs = 2ns, Mem = 5ns, ALU = 3ns$ وبإهمال باقي الأزمنة والمطلوب:

1. احسب زمن الدور الأصغري، والتردد الأعظمي للمعالج



- زمن الدور الأصغري للمعالج يكون محدود ب زمن أبطأ مرحلة في التعليمة يمكن أخذه بحيث يكفي لتنفيذ جميع المراحل السابقة. وهنا نلاحظ أن كلا من زمن جلب التعليمة من الذاكرة Mem والذي يرمز له ب F وزمن القراءة والكتابة على الذاكرة Mem والذي يرمز له بـ M كليهما يعتمدان على زمن التأخير $Mem = 5ns$, وبالتالي هو زمن أبطأ مرحلة أي زمن الدور الأصغري هو $T = 5ns$.

- التردد الأعظمي للمعالج يساوي إلى مقلوب زمن الدور الأصغري

$$f = \frac{1}{T} = \frac{1}{5ns} = 200\ MHz$$

2. عند تنفيذ برنامج مؤلف من 6 تعليمات، وبفرض عدم وجود أي نوع من الأعطال, No Hazards احسب كلاً من : $CPUtime, CPI, CC.$

- يعطى قانون حساب عدد الأدوار الكلي لتنفيذ كامل العمليات بالعلاقة التالية

$$CC = P.L\ depth + (IC - 1) + stalls = 5 + 5 + 0 = 10\ cycles$$

- حيث أن: $P.L\ Depth$ هو عمق التوارد ويساوي إلى عدد مراحل تنفيذ التعليمة.
- IC هو عدد التعليمات الكلي للبرنامج.
- Stalls هو عدد التأخيرات الزمنية اللازم إضافتها لمنع حدوث الـ$hazards$ (يلزم إضافتها فقط في حال كان لدينا $hazards$ في تنفيذ التعليمات وهنا لا يوجد لدينا hazards في المسألة).

$$CPI = \frac{CC}{IC} = \frac{10}{6} = 1.66$$

$$CPU_{time} = CC * T = 10 * 5 = 50ns$$

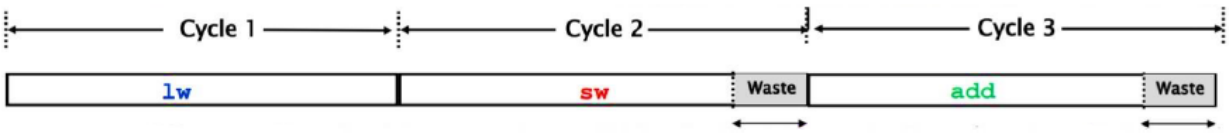
3. عند تنفيذ برنامج مؤلف من 1000000 تعليمة، وبفرض عدم وجود أي نوع من الأعطال، احسب الـCPI.

$$CPI = \frac{5+(IC-1)+stalls}{IC} = 1000004/1000000 = 1.000004 \approx 1$$

- ملاحظة: كلما كانت الCPI قريبة من 1 كلما كان أداء المعالج أفضل ونلاحظ أنه عندما يكون عدد التعليمات كبير نسبيا يكون أداء المعالج أفضل أي أننا نستفاد من تقنية التوارد بشكل أكبر.

- تمرين 2: بفرض تم تنفيذ 3 تعليمات هي lw, sw, add على ثلاثة معالجات: أحادي الدور SC فيه $T = 5ns$ ، ومتعدد الأدوار MC فيه $T = 1ns$ ، ومتوارد PL ذي 5 مراحل فيه $T = 1ns$ وبفرض عدم وجود أية أعطال، ما هو زمن التنفيذ بالحالات الثلاثة؟

1. في مسار المعطيات أحادي الدور $Single\ Cycle\ Data\ Path$

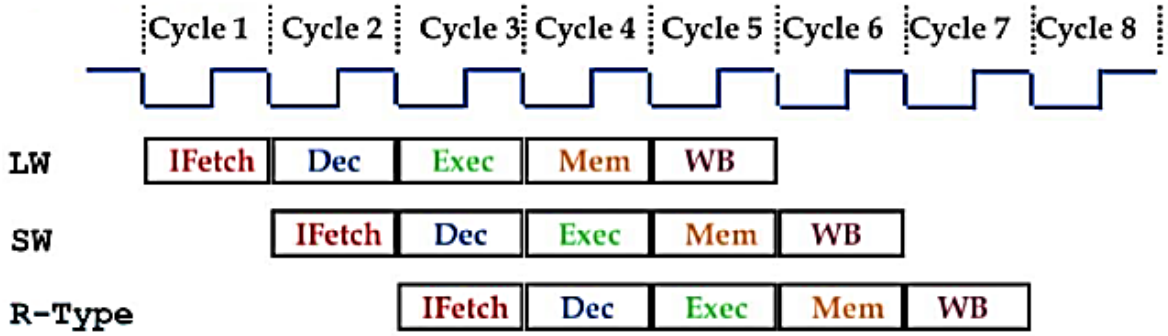


$$CPU_{time} = CC * T = 3 * 5 = 15ns$$

2. في مسار المعطيات متعدد الأدوار *Multiple Cycle Data Path*

$$CPU_{time} = CC * T = (5 + 4 + 4) * 1 = 13ns$$

3. في مسار المعطيات المتوارد *PipeLined Data Path*

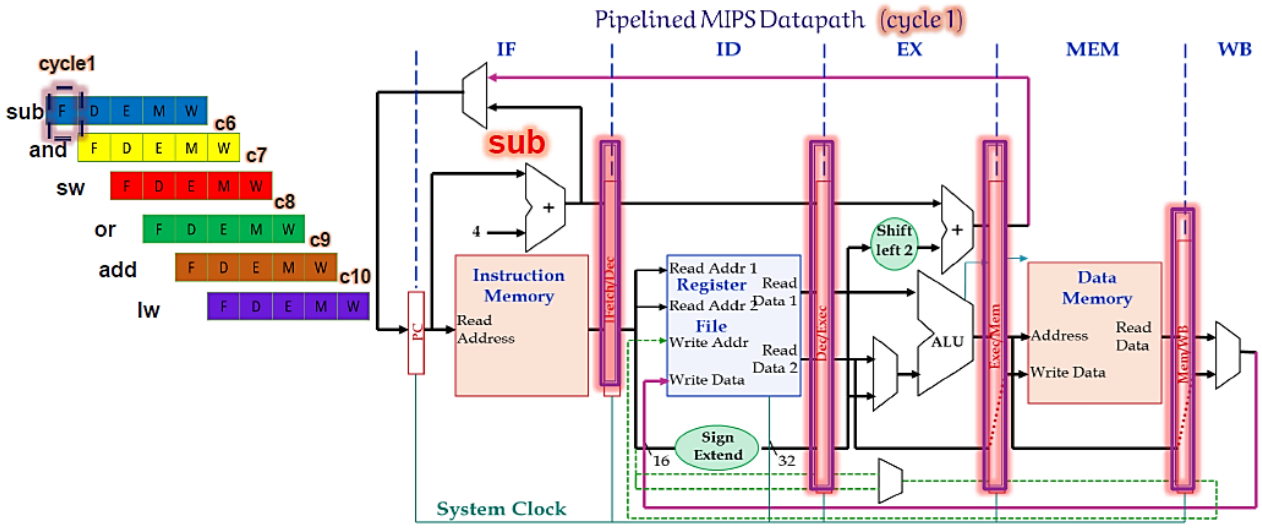


$$CPU_{time} = CC * T = (5 + 2) * 1 = 7ns$$

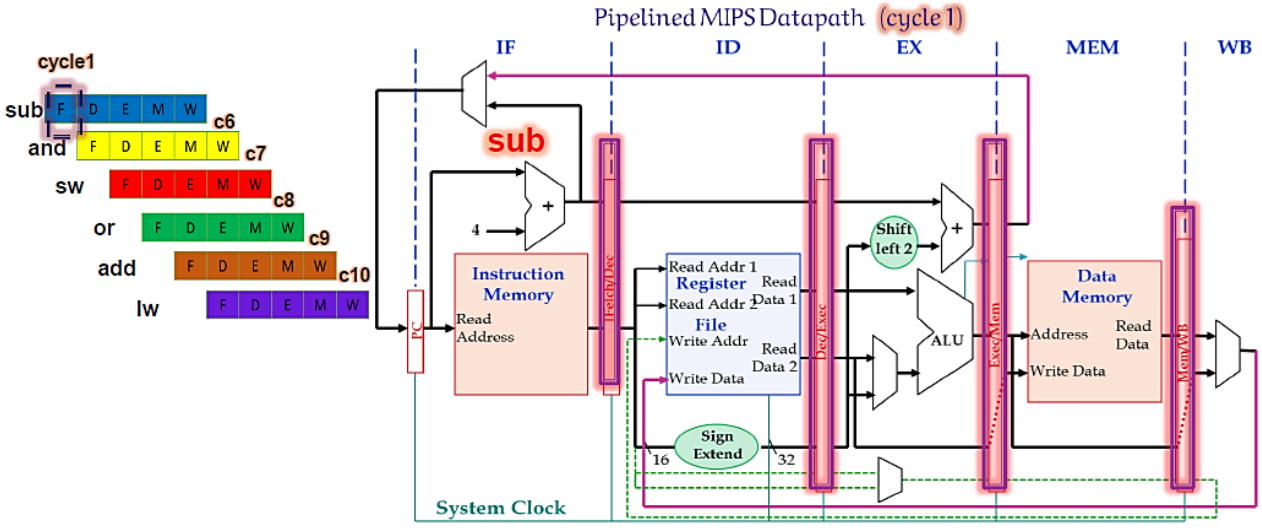
توضيح

1. في مسار المعطيات أحادي الدور يكون الدور بحسب أطول تعليمة سوف تنفذ بحيث كل تعليمة سوف تنتظر هذا الزمن بغض النظر عن طولها. علما أنه لا يمكن الانتقال للتعليمة التالية إلا عند الانتهاء من تنفيذ التعليمة الحالية. ويكون عدد الأدوار الكلي هو أطول تعليمة × عدد التعليمات.
2. في مسار المعطيات متعدد الأدوار تم تقسيم التعليمات إلى مراحل بحيث أصبح الدور هو زمن تنفيذ المرحلة الواحدة ولا داعي أن تنتظر التعليمة أكبر دور في التعليمات. علما أنه لا يمكن الانتقال للتعليمة التالية إلا عند الانتهاء من تنفيذ التعليمة الحالية. ويكون عدد الأدوار الكلي هو مجموع مراحل كل تعليمة.
3. في مسار المعطيات المتوارد تم تقسيم عملية المعالجة إلى مراحل بحيث عند الانتهاء من مرحلة F مثلاً في تعليمة معينة يمكن البدء في نفس المرحلة للتعليمة القادمة دون الانتظار الانتهاء تنفيذ التعليمة الأولى. وهنا يمكن الانتقال للتعليمة التالية حال الانتهاء من مرحلة معينة، إذ يمكن بدء التعليمة التالية عند هذه المرحلة. ويكون عدد الأدوار الكلي هو عمق التعليمة + عدد التعليمات - 1

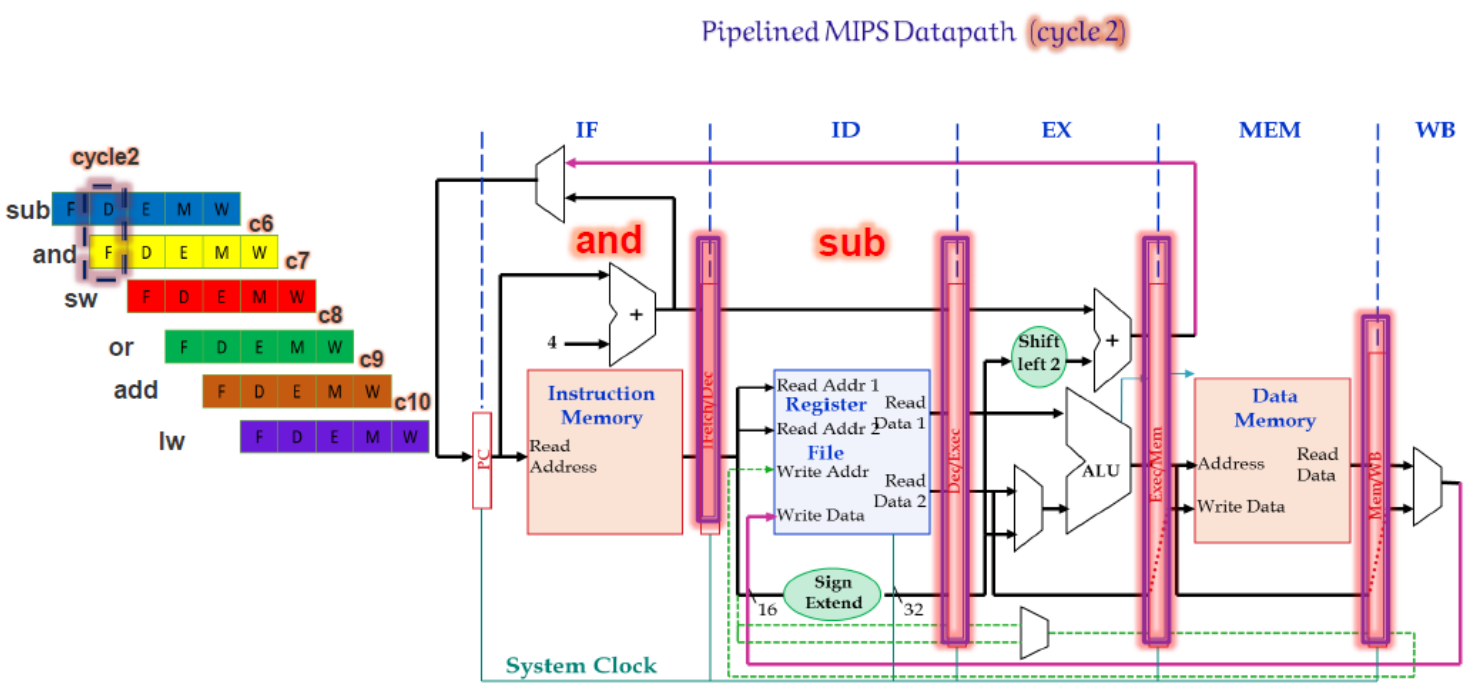
- **تمرين 3** بفرض تم تنفيذ 6 تعليمات هي $sub \rightarrow and \rightarrow sw \rightarrow or \rightarrow add \rightarrow lw$ على المعالج المتوارد التالي، وبفرض عدم وجود أية أعطال، حدد توزع التعليمات على المخطط المرفق في أول 5 أدوار من بدء التنفيذ، ثم حدد في أي دور ساعة يتم حفظ ناتج كل من التعليمتين or و lw في ملف السجلات؟



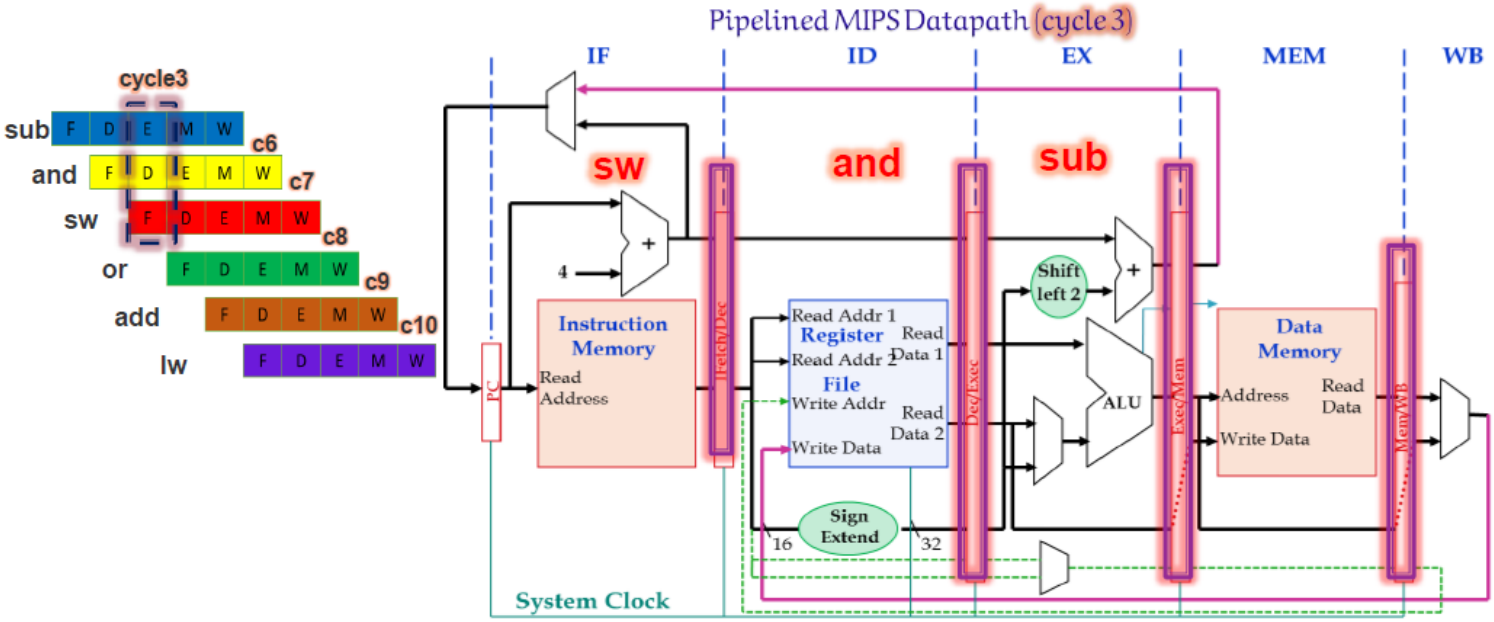
ملاحظة تم تقسيم عملية المعالجة إلى مراحل $(F \ / \ D \ / \ E \ / \ M \ / \ W)$ المرمزة فوق كل قسم في الرسمة، عند طلب توزع التعليمات في دور ما نبحث ضمن الدور عن المراحل التي تتم معالجتها في هذا الدور. وبما أن تنفيذ التعليمة يتم على عدة أدوار لذا يلزم سجلات بينية بين المراحل لحفظ المعاملات والمعطيات وإشارات التحكم الخاصة بكل تعليمة مما تحتاجه المراحل التالية.



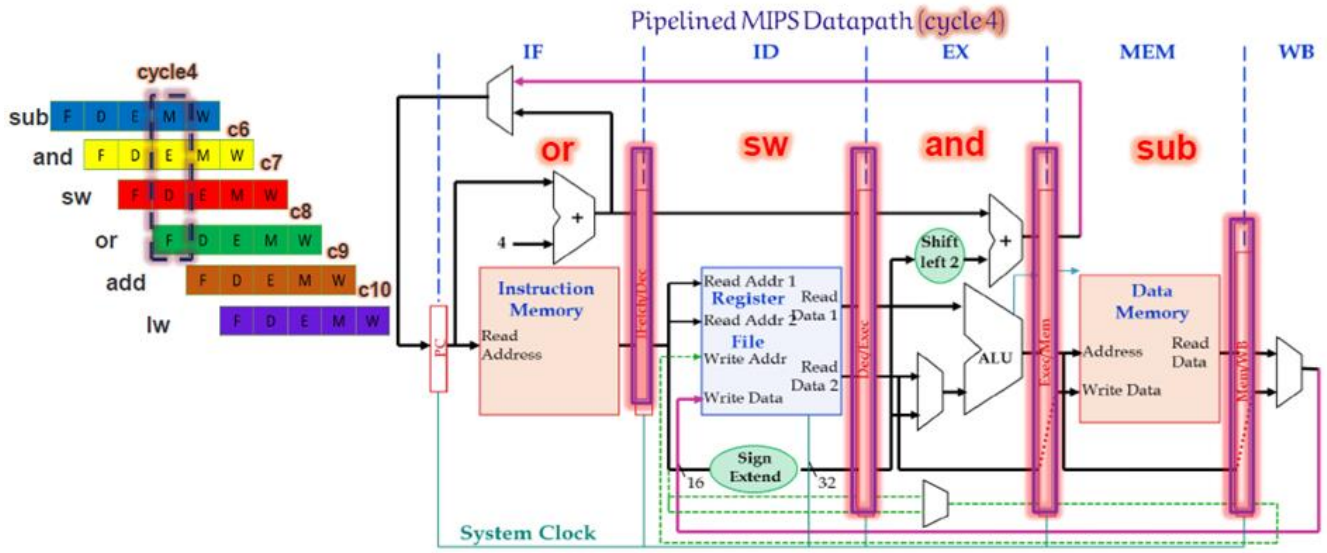
في الدور الأول تكون عملية sub في طور جلب التعليمة IF.



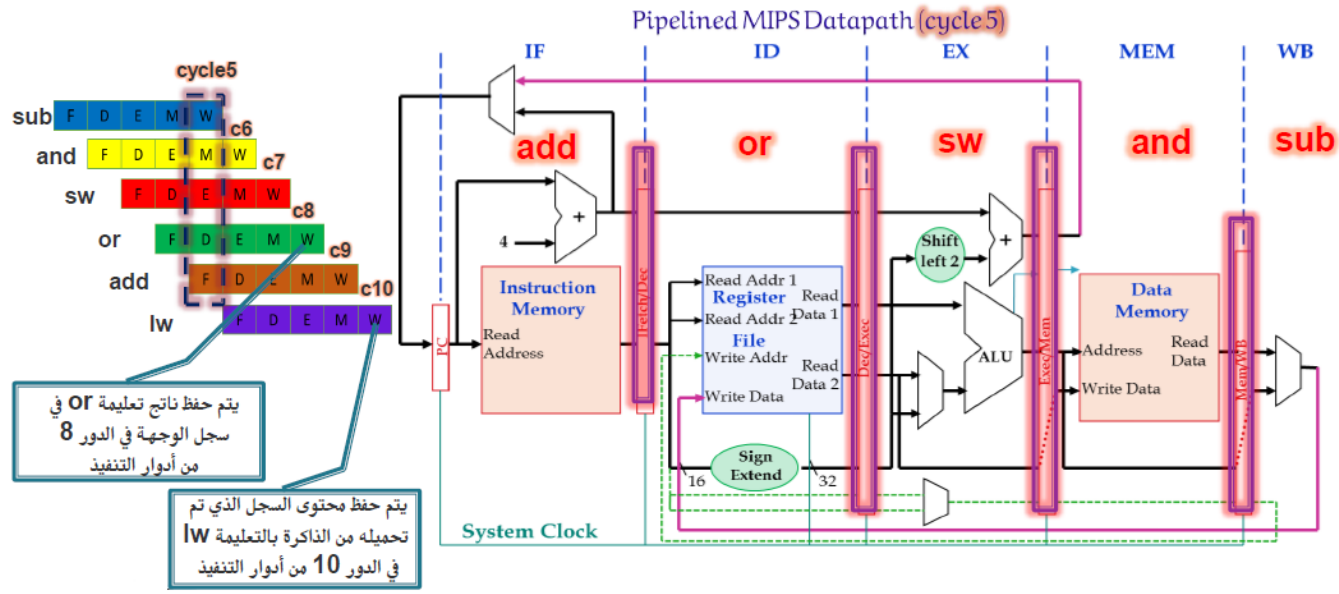
في الدور الثاني تنتقل عملية sub إلى مرحلة فك التشفير ID عن طريق السجل البيني IF/ID الواصل بين مرحلة جلب التعليمة ومرحلة فك التشفير وفي ذات الوقت تبدأ عملية and في مرحلة جلب التعليمة IF.



في الدور الثالث تنتقل عملية sub إلى مرحلة التنفيذ EX عن طريق السجل البيني ID/EX الواصل بين مرحلة فك التشفير ومرحلة التنفيذ وفي نفس الوقت تنتقل عملية and إلى مرحلة فك التشفير ID وذلك عن طريق السجل البيني IF/ID الواصل بين مرحلة جلب التعليمة ومرحلة فك التشفير في حين أن عملية sw تبدأ بمرحلة الجلب IF.



في الدور الرابع تنتقل عملية sub إلى مرحلة القراءة والكتابة على الذاكرة Mem عن طريق السجل البيني EX/MEM الواصل بين مرحلة التنفيذ ومرحلة القراءة والكتابة على الذاكرة وفي نفس الوقت تنتقل عملية and إلى مرحلة التنفيذ EX عن طريق السجل البيني ID/EX الواصل بين مرحلة فك التشفير ومرحلة التنفيذ وفي ذات الوقت تنتقل عملية sw إلى مرحلة فك التشفير ID وذلك عن طريق السجل البيني IF/ID الواصل بين مرحلة جلب التعليمة ومرحلة فك التشفير في حين أن عملية sw تبدأ بمرحلة الجلب IF.



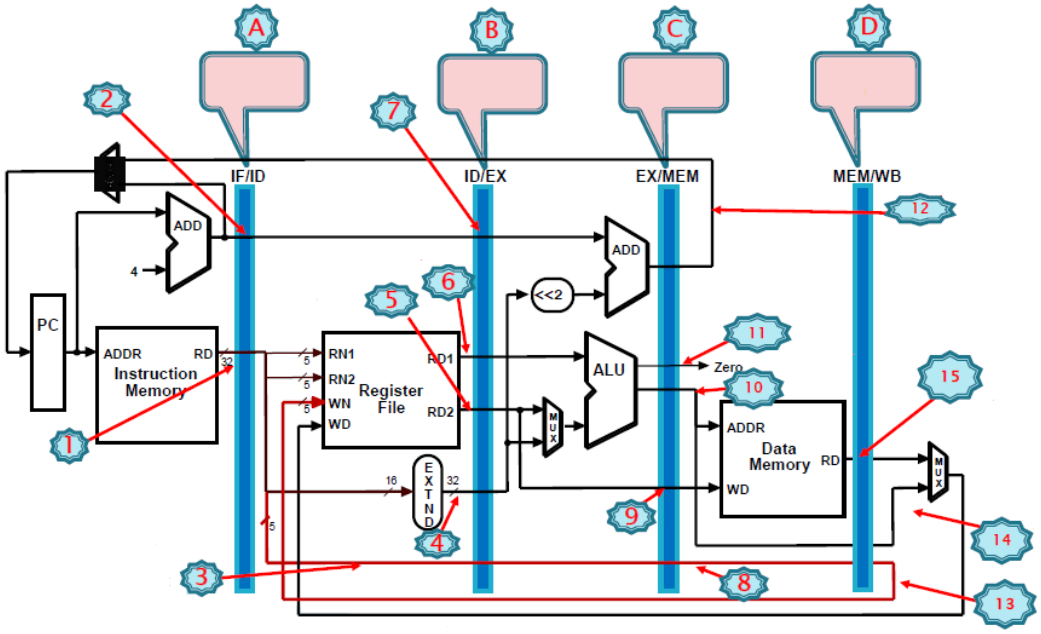
أما في الدور الخامس تنتقل عملية sub إلى طور العودة للكتابة على ملف السجلات WB (والذي من خلاله يتم حفظ نتيجة العملية بعد التنفيذ) بواسطة السجل البيني MEM/WB الواصل بين مرحلة القراءة والكتابة على الذاكرة ومرحلة العودة للكتابة. وفي أثناء ذلك تنتقل عملية and إلى مرحلة القراءة والكتابة على الذاكرة Mem عن طريق السجل البيني EX/MEM الواصل بين مرحلة التنفيذ ومرحلة القراءة والكتابة على الذاكرة وفي نفس الوقت تنتقل عملية sw إلى مرحلة التنفيذ EX عن طريق السجل البيني ID/EX الواصل بين مرحلة فك التشفير ومرحلة التنفيذ وفي ذات الوقت تنتقل

عملية or إلى مرحلة فك التشفير ID وذلك عن طريق السجل البيني IF/ID الواصل بين مرحلة جلب التعليمة ومرحلة فك التشفير في حين أن عملية add تبدأ بمرحلة الجلب IF.

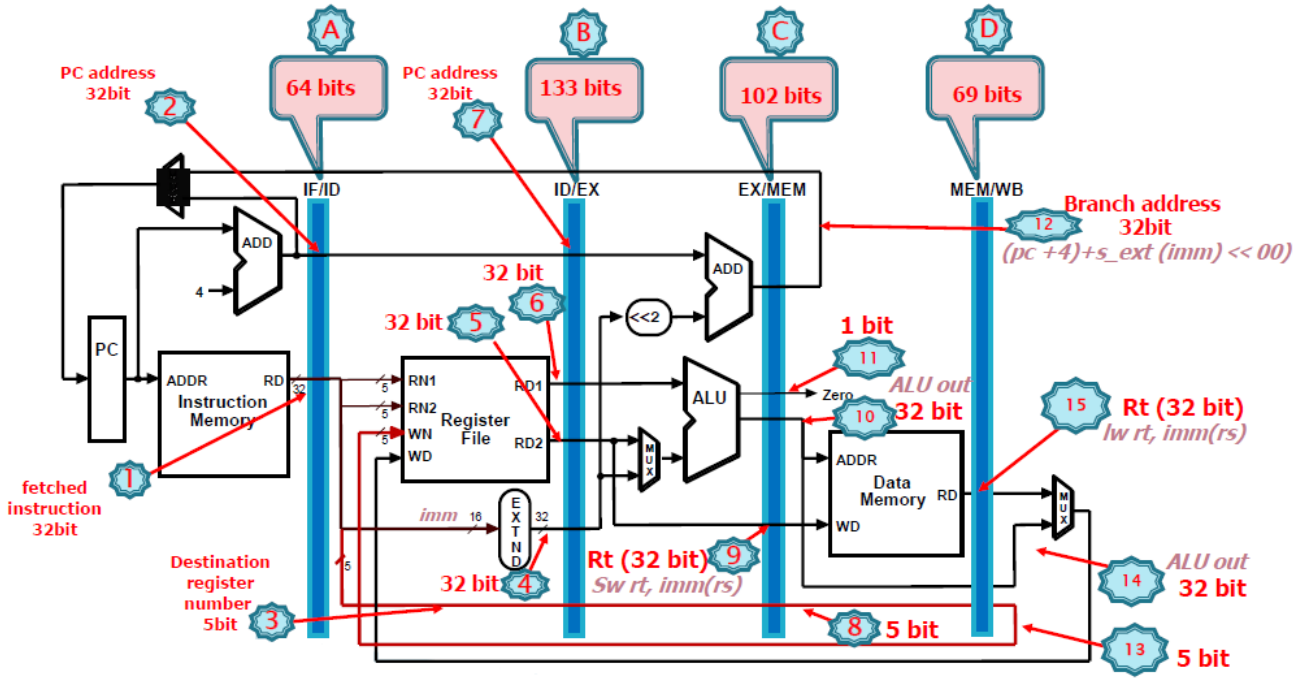
للإجابة عن سؤال تحديد دور الساعة الذي يتم فيه حفظ ناتج كل من التعليمتين or و lw في ملف السجلات، نقوم بالبحث في الرسمة السابقة (رسم مخطط الدور الخامس) فنلاحظ أنه يتم حفظ ناتج تعليمة or في سجل الوجهة في الدور 8 من أدوار التنفيذ. في حين أنه يتم حفظ محتوى السجل الذي تم تحميله من الذاكرة بالتعليمة lw في الدور 10 من أدوار التنفيذ.

تمرين 4: (غير مطلوب في القسم العملي لكن يفيد في فهم الأفكار النظرية لبحث التوارد)

- حدد عرض السجلات البينية IF/ID, ID/EX, EX/MEM, MEM/WB في الشكل التالي:



الحل:



توضيح عملية الحل:

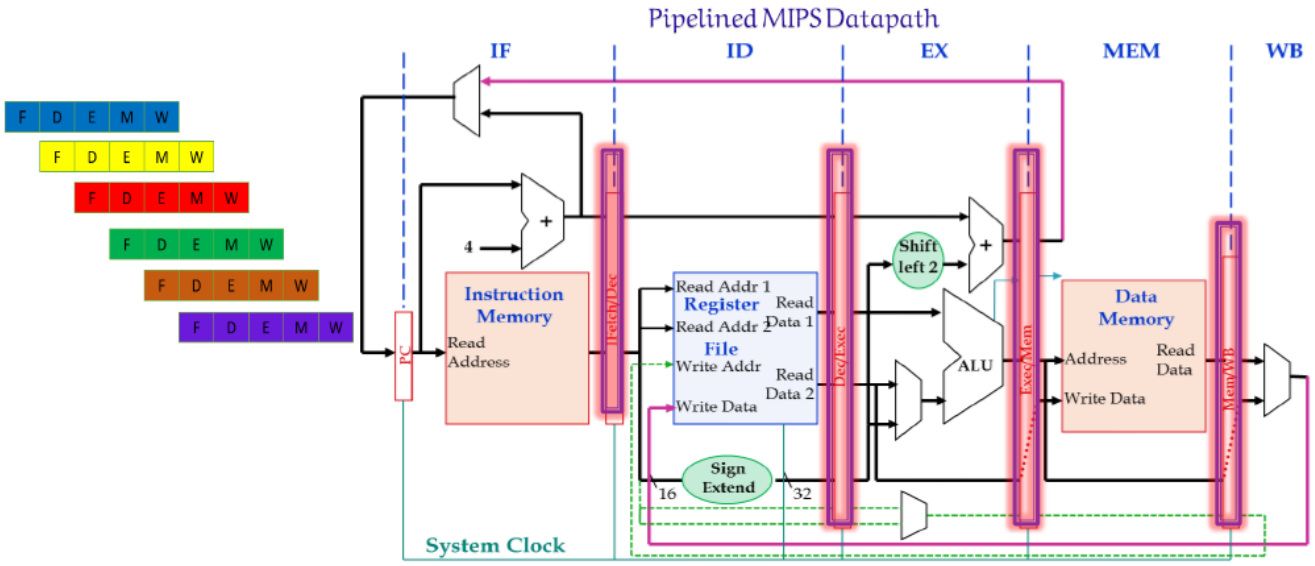
- يدخل إلى السجل البيني IF/ID (الشكل A) مدخلين هما: الرقم 1 يمثل مسار التعليمة بعد الجلب والذي يخرج من المخرج RD من ذاكرة التعليمات ويبلغ حجمها 32 بت. أما الرقم 2 فـ يمثل مسار عنوان عداد البرنامج PCبعد عملية الزيادة بمقدار $(PC = PC + 4)$من قبل ال ALUويكون حجم محتوى هذا العنوان حوالي 32 بت. وبجمع القيم السابقة بالتالي يكون عرض السجل البيني IF/ID هو 64بت.

- يدخل إلى السجل البيني ID/EX(الشكل B) خمسة مدخلات هم: الرقم 3 يمثل مسار رقم سجل التعليمة rdالعائدة بعد عملية ال $WB\ (Write-Back)$ ويبلغ حجمها 5 بت. الرقم 4 يمثل مسار القيمة الفورية immمن ذاكرة التعليمات وذلك بعد اجراء عملية $extend$عليها عند تطبيق التعليمات من النوع $I-Type$ فيكون حجمها قبل العملية هو 16بت أما بعد العملية (وهو الحجم المطلوب للسجل البيني) فيكون حجمها 32بت. الرقم 5 يمثل مسار المخرج $RD2$ من ملف السجلات والذي يحمل سجل المعطيات rt عند تطبيق التعليمات من النوع $I-Type$ ويكون حجمه 32 بت. الرقم 6 يمثل مسار المخرج $RD1$ من ملف السجلات والذي يحمل سجل المعطيات rs عند تطبيق التعليمات من النوع $R-Type$ ويكون حجمه 32 بت. الرقم 7 يمثل مسار عنوان عداد البرنامج PC ويكون حجمه 32 بت. وبجمع القيم السابقة بالتالي يكون عرض السجل البيني ID/EX هو 133 بت.

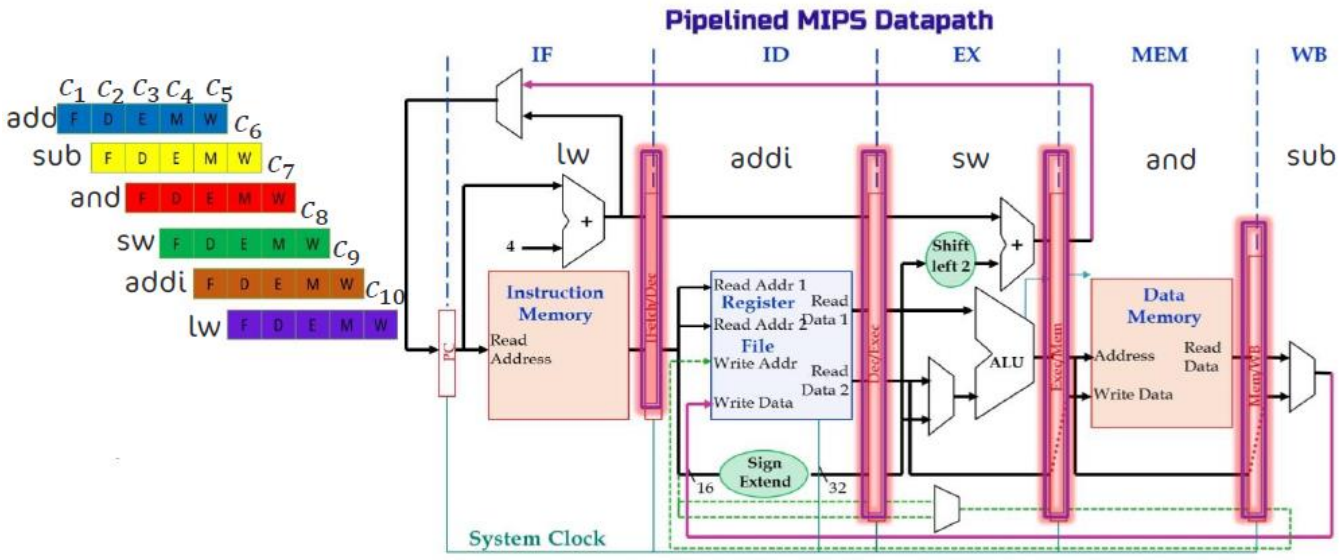
- يدخل إلى السجل البيني EX/MEM (الشكل C) خمسة مدخلات هم: الرقم 8 يمثل مسار رقم سجل التعليمة rdالعائدة بعد عملية ال WB (Write-Back) ويبلغ حجمها 5 بت. الرقم 9 يمثل مسار سجل المعطيات rt عند تطبيق تعليمة الsw وتكون واصلة إلى الذاكرة Mem ويكون حجمها 32 بت. الرقم 10 يمثل مسار المحتوى الناتج ALUout بعد عملية التنفيذ في ال ALU ويكون حجمها 32 بت. أما الرقم 11 يمثل البت الصفري zero وهو يتألف من بت وحيد يفيد في تعليمات التفرع الشرطي بحيث يعطي قيمة 1 عند وجود عملية تفرع بعد تحقق عملية المقارنة بين السجلات وقيمة 0 خلاف ذلك. الرقم 12 يمثل مسار عنوان عداد البرنامج PC بعد عملية التفرع بحيث يحسب عن طريق إضافة مقدار 4 للعنوان وجمعه مع القيمة الفورية imm بعد تطبيق عملية ال extend بالإضافة إلى إزاحة الناتج بمقدار صفرين نحو اليسار، يكون حجم محتوى عداد عنوان البرنامج بعد تطبيق كل هذه العمليات هو 32 بت. وبجمع القيم السابقة نجد أن عرض السجل البيني EX/MEM هو 102 بت.

- يدخل إلى السجل البيني MEM/WB (الشكل D) ثلاثة مدخلات هم: الرقم 13 يمثل مسار رقم سجل التعليمة rd العائدة بعد عملية ال $WB\ (Write-Back)$ ويبلغ حجمها 5 بت. الرقم 14 يمثل مسار المحتوى الناتج $ALUout$ بعد عملية التنفيذ في ال ALU ويكون حجمها 32 بت. الرقم 15 يمثل مسار المخرج RD من الذاكرة Mem والذي يحمل محتوى السجل rt أثناء تطبيق عملية الـlw. ويكون حجم محتوى هذا السجل هو 32 بت. عند جمع القيم السابقة نجد أن عرض السجل البيني MEM/WB هو 69 بت.

- **وظيفة 1:** بفرض تم تنفيذ 6 تعليمات هي $add \rightarrow sub \rightarrow and \rightarrow sw \rightarrow addi \rightarrow lw$ على المعالج المتوارد التالي، وبفرض عدم وجود أية أعطال، حدد توزع التعليمات على المخطط المرفق في الدور السادس من بدء التنفيذ، ثم حدد في أي ساعة يتم حفظ ناتج التعليمات $sub, addi, lw$ في ملف السجلات؟



الحل:



بنفس طريقة حل التمرين الثالث ولتحديد دور الساعة الذي يتم فيه حفظ ناتج التعليمات $sub, addi, lw$في ملف السجلات، نقوم بالبحث في رسم مخطط الدور السادس (وذلك بعد توزيع كل التعليمات على المعالج المتوارد) عن المرحلة w في التعليمات المطلوبة فنلاحظ أنه سيتم حفظ التعليمة sub في الدور السادس والتعليمة $addi$ في الدور التاسع والتعليمة lw في الدور العاشر.

الأعطال Hazards

- مراحل معالج MIPS المتوارد الخمسة:

1. جلب التعليمة Fetch
2. فك تشفير التعليمة Decode
3. التنفيذ في وحدة الحساب والمنطق Execution
4. القراءة والكتابة على الذاكرة Memory
5. العودة للكتابة على ملف السجلات Write-Back

أنواع الأعطال

1. الأعطال البنيوية Structural Hazards

تحدث نتيجة التنازع على نفس الموارد في المعالج (مثل وجود ذاكرة وحيدة، أو ALU وحيدة.. الخ)، مثل تعليمة تقوم بعملية جلب تعليمة من الذاكرة وتعليمة أخرى تقوم بالتخزين على الذاكرة في نفس الدور حيث لا يمكن استخدام نفس المورد في الوقت ذاته لتعليمتين مختلفتين. ولحل العطل البنيوي نلجأ إلى الانتظار حتى الانتهاء من استخدام المورد أو نلجأ إلى شراء مورد جديد.

ملاحظة سنفترض في مسائل العملي أن الأعطال البنيوية محلولة بتوفر الموارد اللازمة، مثلاً يمكن حل مشكلة التنازع على الذاكرة بتوفير خابيتين للتعليمات وللمعطيات $D_Cache, I_Cache \cdots$

2. أعطال المعطيات Data Hazards

ناتجة عن ترابط المعطيات أي عندما تحتاج تعليمة لمعطيات ناتجة عن تعليمة سابقة لا تزال موجودة في قناة التوارد PL (مثل طلب قراءة قيمة سجل تم التعديل عليه بتعليمة سابقة قبل تحديث القيمة).

3. أعطال التحكم Control Hazards

ناتجة عن تعليمات القفز والتفرع $(j, beq, bne, \dots)$.

ملاحظة سنتناول في هذه المحاضرة فقط أعطال المعطيات باعتبار أن الأعطال البنيوية محلولة سابقا, أما أعطال التحكم ف سنبدأ بها في المحاضرة القادمة.

"Everything you can imagine is real."

—Pablo Picasso

أعطال المعطيات Data Hazards

- تحدث عندما تحتاج تعليمة لمعطيات ناتجة عن تعليمة سابقة لا تزال موجودة في قناة التوارد PL .
- الحلول الممكنة:
  - التأخير الزمني Stalls / إضافة فقاعات Bubbles (حل عتادي يحتاج إضافة دارة كشف الأعطال $Hazarrd\ Detection\ Unit$).
  - إضافة دارة إحالة $Bypassing\ /\ Forwarding\ Unit$، للحصول على المعطيات دون انتظار الكتابة النهائية على الوجهة (حل عتادي يحتاج أيضاً إضافة دارة كشف الأعطالHazard Detection Unit).
  - إعادة ترتيب البرنامج عن طريق المترجم Compiler بحيث يتم المباعدة بين التعليمات المترابطة، وإضافة NOP عند الحاجة (حل برمجي).

ملاحظة: يتم تمثيل الNOP في لغة تجميع الMIPS على الشكل التالي:

In $MIPS$: $NOP = \text{“}no\ operation\text{”} = 00 \dots 0\ (32bits) = sll\ \$0,\$0,0\ \#SLL{:}\ shift\ left\ logical$

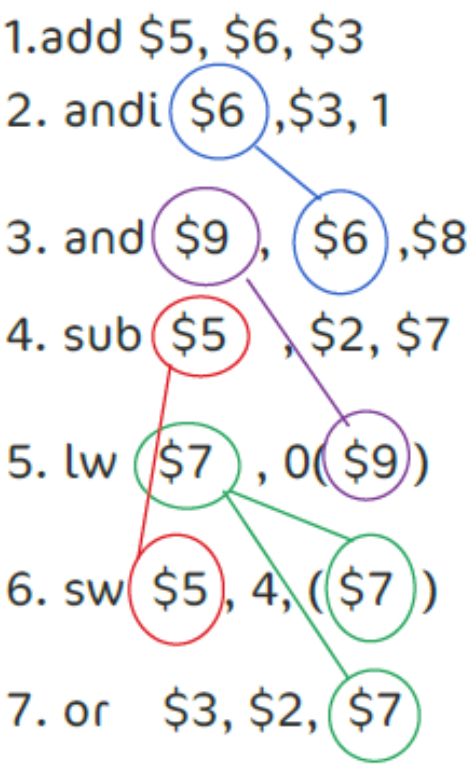
- **تمرين 5:** لدينا البرنامج التالي ينفذ على معالج MIPS متوارد، حدد ترابطات المعطيات الموجودة في البرنامج، ثم مثّل أدوار التنفيذ باستخدام طريقة التأخير الزمني Stall (إضافة فقاعات Bubble) لمعالجة أعطال المعطيات بفرض عدم وجود أعطال بنيوية، وبفرض إمكانية النفاذ المزدوج لملف السجلات (إمكانية الكتابة والقراءة بنفس الدور).

1. $sub\ \$5,\$6,\$3$
2. $andi\ \$6,\$3\ ,1$
3. $and\ \$9,\$6\ ,\$8$
4. $add\ \$5\ ,\$2\ ,\$7$
5. $lw\ \$7\ ,0(\ \$9\ )$
6. $sw\ \$5\ ,4\ (\ \$7\ )$
7. $or\ \$3\ ,\$2\ ,\$7$

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

1. بفرض أن قراءة السجلات تتم في المرحلة D

- نحدد الترابطات بين أسطر التعليمات عن طريق البحث عن أي تعليمة تقوم بقراءة سجل بعد تعليمة قامت بالكتابة على نفس السجل (read after write RAW) فنلاحظ أنه في أول تعليمة لا يوجد قبلها كتابة على أي سجل فلا يوجد ترابط. في التعليمة الثالثة نلاحظ وجود ترابط بينها وبين سابقتها حيث في التعليمة الثانية يتم الكتابة على السجل $6 وبنفس الوقت تتم قراءة قيمة السجل في التعليمة الثالثة مما قد يؤدي لحدوث أخطاء لدى عملية القراءة.



- يوجد أيضا الترابط بين التعليمتين السادسة والرابعة حيث تتم عملية القراءة من محتوى السجل $5 في التعليمة السادسة أثناء انتهاء تنفيذ عملية الكتابة على نفس السجل في التعليمة الرابعة. كما يوجد ترابط بين التعليمتين الخامسة والثالثة حيث تتم عملية القراءة من محتوى السجل $9 في الذاكرة من العنوان 0 في التعليمة الخامسة أثناء انتهاء تنفيذ الكتابة على نفس السجل في التعليمة الثالثة. كما يوجد ترابط بين التعليمات الخامسة والسادسة والسابعة حيث تتم عملية القراءة من محتوى السجل $7 في الذاكرة من العنوان 4 في التعليمة السادسة أثناء انتهاء تنفيذ الكتابة على نفس السجل في التعليمة الخامسة. وتتم عملية القراءة من السجل $7 من أجل تنفيذ العملية المنطقية or في التعليمة السابعة أثناء انتهاء تنفيذ الكتابة على نفس السجل في التعليمة الخامسة.

ملاحظة لم نأخذ ترابط بين التعليمة السادسة والتعليمة الأولى (على الرغم من وجود عملية قراءة ل $5 في السادسة وعملية كتابة على $5 في الأولى) والسبب هو أننا نبحث عن آخر تعديل على قيمة السجل ولا نهتم للتعديلات السابقة فقيمة $5 تعدلت في التعليمات الأولى والرابعة وقيمتها التي تهمنا بعد آخر تعديل فالترابط يكون مع آخر تعديل على السجل فقط.

كيفية تعبئة الجدول

- **عند تعبئة الجدول في حال وجود ترابط بين التعليمة المراد تعبئتها مع تعليمات سابقة ننظر إلى المرحلة D في تعليمتنا وإلى المرحلة في التعليمة المترابطة معها ونناقش الحالات الثلاثة التالية:**

1. تمت عملية W في دور قبل الدور الذي يمكن وضع D فيه: نضع D ونكمل بشكل طبيعي.
2. تتم عملية W في نفس الدور الذي يمكن وضع D فيه: في حال النفاذ المزدوج نضع D بشكل طبيعي وفي حال عدم وجود نفاذ نضع فقاعة ثم D..
3. تتم عملية W في دور بعد الدور الذي يمكن وضع D فيه: نترك فقاعات إلى أن نصل إلى الدور W ونضع D بحيث تكون D تحت W مباشرة وذلك في حال النفاذ المزدوج وفي حال عدم وجوده نترك فقاعة زيادة.

فيصبح الجدول على الشكل التالي

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
F	D	E	M	W											
	F	D	E	M	W										
		F			D	E	M	W							
					F	D	E	M	W						
						F		D	E	M	W				
								F			D	E	M	W	
											F	D	E	M	W

$$CPI = \frac{5 + (IC - 1) + stalls}{IC} = \frac{5 + 6 + 5}{7} = \frac{16}{7} = 2.3$$

ملاحظة بما أن التمرين ينص على إمكانية النفاذ المزدوج إلى ملف السجلات أمكننا وضع المرحلة D في الدور السادس مع المرحلة W على نفس السجل وفي نفس الدور وإذا لم ينص التمرين على النفاذ المزدوج يجب وضع فقاعة ثالثة والانتظار للانتهاء من W و وضع D في الدور السابع .

2. أعد الحل بفرض أن قراءة السجلات تتم في المرحلة E

- بنفس الطريقة السابقة تماما ولكن من أجل المرحلة E نجد أن:

1. sub $5,$6,$3
2. andi $6, $3 , 1
3. and $9, $6, $8
4. add $5 , $2 , $7
5. lw $7 ,0($9
6. sw $5 , 4 ($7)
7. or $3 , $2 , $7

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
F	D	E	M	W												
	F	D	E	M	W											
		F	D	*	*	E	M	W								
			F	*	*	D	E	M	W							
						F	D	*	E	M	W					
							F	*	D	*	*	E	M	W		
									F	*	*	D	E	M	W	

ملاحظة:
لا يمكن هنا إجراء W و E على نفس السجل في نفس الدور لأن خاصية النفاذ المزدوج تنص على إمكانية الكتابة على ملف السجلات في نصف الدور ثم القراءة منه في نصف الدور، ولا تنص على إمكانية إجراء E في نصف الدور!

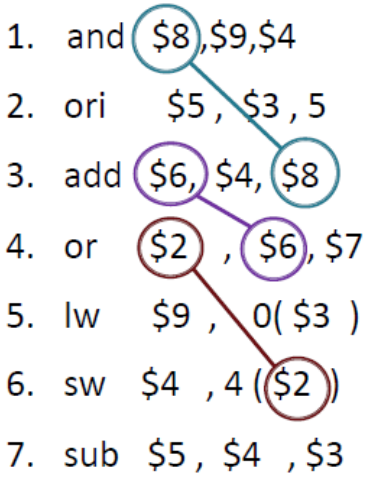
$$CPI = \frac{\{5+(IC-1)+stalls\}}{IC} = \frac{\{5+6+5\}}{7} = \frac{16}{7} = 2.3$$

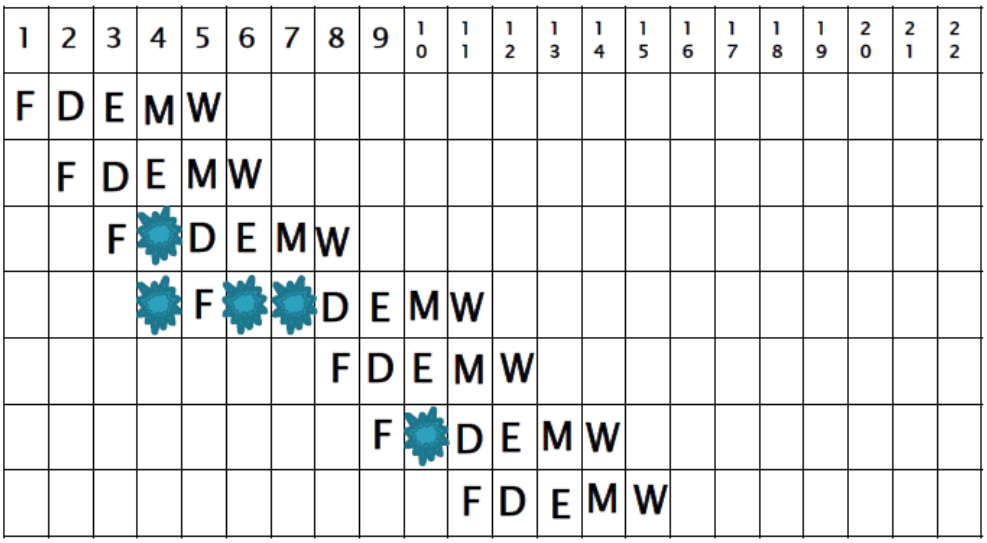
- تمرين 6: (للتدريب): بفرض لدينا البرنامج التالي ينفذ على معالج MIPS متوارد، حدد ترابطات المعطيات الموجودة في البرنامج، ثم مثّل أدوار التنفيذ بفرض استخدام طريقة التأخير الزمني Stall (إضافة فقاعات Bubble) لمعالجة أعطال المعطيات بفرض عدم وجود أعطال بنيوية، وبفرض إمكانية النفاذ المزدوج لملف السجلات (إمكانية الكتابة والقراءة بنفس الدور) .

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22

1. and \$8 , \$9, \$4
2. ori \$5 , \$3 , 5
3. add \$6, \$4, \$8
4. or \$2 , \$6 , \$7
5. lw \$9, 0(\$3)
6. sw \$4 , 4 (\$2)
7. sub \$5 , \$4 , \$3

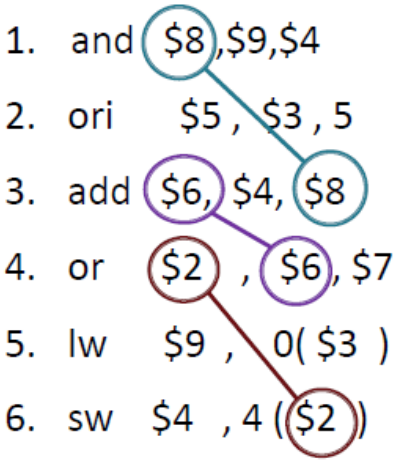
1. بفرض أن قراءة السجلات تتم في المرحلة D

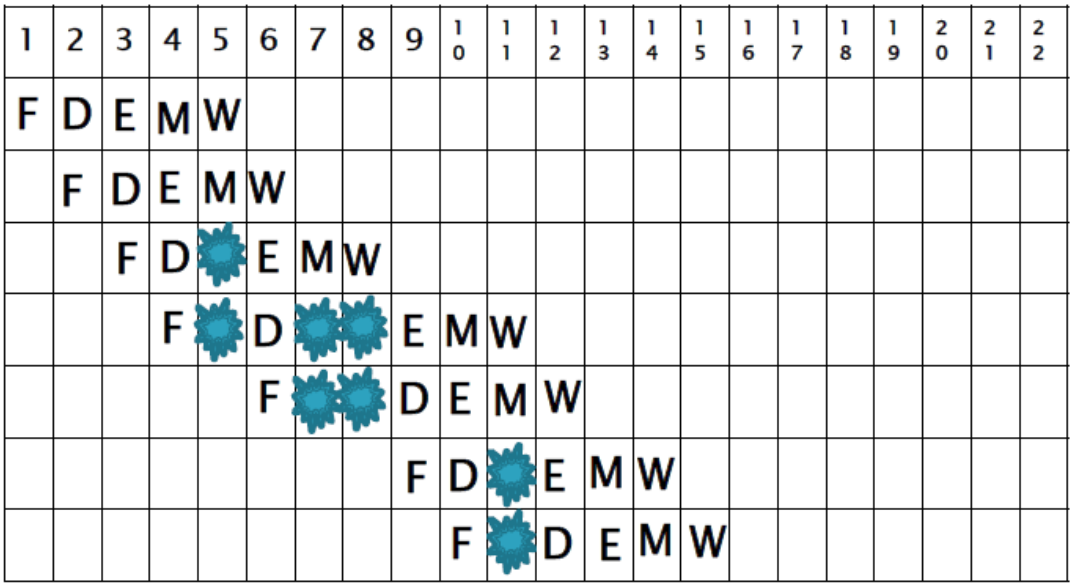




$$CPI = \frac{\{5+(IC-1)+stalls\}}{IC} = \frac{\{5+6+4\}}{7} = \frac{15}{7} = 2.14$$

2. أعد حل التمرين السابق بفرض أن قراءة السجلات تتم في المرحلة E





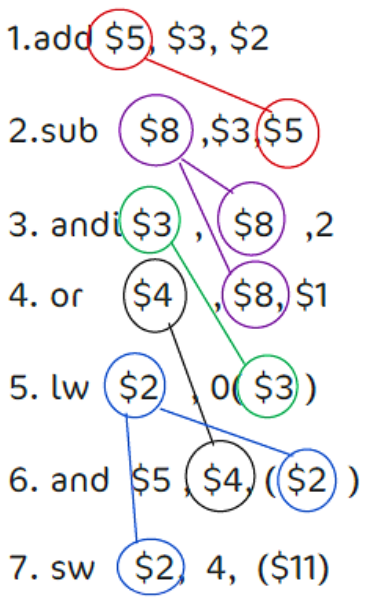
$$CPI = \frac{\{5+(IC-1)+stalls\}}{IC} = \frac{\{5+6+4\}}{7} = \frac{15}{7} = 2.14$$

"Don't limit your challenges. Challenge your limits."

- **وظيفة 2 (للتدريب):** حدد ترابطات المعطيات ثم مثّل أدوار تنفيذ البرنامج التالي على المعالج المتوارد بخمس مراحل، واحسب CPI (بفرض عدم وجود أعطال بنيوية وعدم وجود دارة إحالة) وبفرض أن قراءة السجلات تتم في المرحلة E.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1. add $5,$3,$2																				
2. sub $8, $3, $5																				
3. andi $3, $8, 2																				
4. or $4 , $8 ,$1																				
5. lw $2 ,0 ($3)																				
6. and $5, $4, $2																				
7. sw $2 , 4 ($11)																				

الحل:



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1.add $5, $3, $2	F	D	E	M	W															
2.sub $8 ,$3 ,$5		F			D	E	M	W												
3. andi $3 , $8 ,2					F			D	E	M	W									
4. or $4 , $8, $1								F	D	E	M	W								
5. lw $2 , 0($3)									F		D	E	M	W						
6. and $5 , $4, ($2)											F			D	E	M	W			
7. sw $2, 4, ($11)														F	D	E	M	W		

$$C = 5 + (7 - 1) + 7 = 5 + 6 + 7 = 18$$

$$CPI = \frac{CC}{IC} = \frac{18}{7} \approx 2.57$$

وبالتالي نجد أنه يوجد حوالي 7 ترابطات بالإضافة إلى 7 أدوار تأخير stalls

وتكون قيمة الـCPIهي$CPI = 18/7 = 2.57$.

انتهت المحاضرة