

<b>Problemas adicionales Tema1 y Tema 2 - 2017</b>
--

**P1.-** Al ejecutar un programa en un procesador que tiene un tiempo de ciclo de 100ns, se obtiene la siguiente distribución de instrucciones.

	Unidad INT	Unidad CF	Memoria LD/ST	Resto
%Instrucciones	50	25	15	10
CPI	2	8	4	4

Se aplican 2 mejoras independientes, una en la unidad de enteros (INT) y otra en la unidad de coma flotante (CF). Al medir el tiempo de ejecución se obtiene que la aceleración global que se consigue es de 2 y se comprueba que la mejora de en la unidad INT ha conseguido que el CPI de este tipo de instrucciones sea 1.

- a) ¿Qué CPI se ha conseguido para las instrucciones de CF? ¿Qué aceleración parcial se ha conseguido con esta mejora?
- b) Aplicando la ley de Amdahl ¿Qué valores corresponde a los parámetros  $F_{m1}$ ,  $A_{m1}$  asociados a la mejora en la unidad INT y  $F_{m2}$ ,  $A_{m2}$  asociados a la mejora en CF?
- c) ¿Qué porcentaje del tiempo de ejecución del sistema inicial no se ha mejorado?

## Problemas adicionales Tema1 y Tema 2 - 2017

### Solución

a)  $T_{\text{ejecución sin mejoras}} = NI \cdot (0,50 \times 2 + 0,25 \times 8 + 0,15 \times 4 + 0,10 \times 4) = NI(1 + 2 + 0,6 + 0,4) = 4 \times NI \text{ ciclos}$

Como  $Ag = 2$  entonces el  $T_{\text{ejecución mejorado}}$  es la mitad  
 $CPI_{INT\text{mejorado}} = 1$ .

$$T_{\text{ejecución mejorado}} = 2 \times NI = NI \cdot (0,50 \times 1 + 0,25 \times CPI_{CF\text{mejorado}} + 0,15 \times 4 + 0,10 \times 4)$$

$$2 = 0,50 \times 1 + 0,25 \times CPI_{CF} + 0,15 \times 4 + 0,10 \times 4$$

$$2 - 0,5 - 1 = 0,25 \times CPI_{CF}$$

$$CPI_{CF} = 2$$

b)  $Am_1 = 2$ ;  $Fm_1 = (0,5 \times 2)/4$ ;  $Fm_1 = 0,25$   
 $Am_2 = 4$ ,  $Fm_2 = (0,25 \times 8)/4 = 0,5$

$$Ag = 2 = 1 / ((1 - fm_1 - fm_2) + fm_1 / Am_1 + fm_2 / Am_2)$$

$$= 1 / (0,25 + 0,25/2 + 0,5/4) = 1/0,5 = 2$$

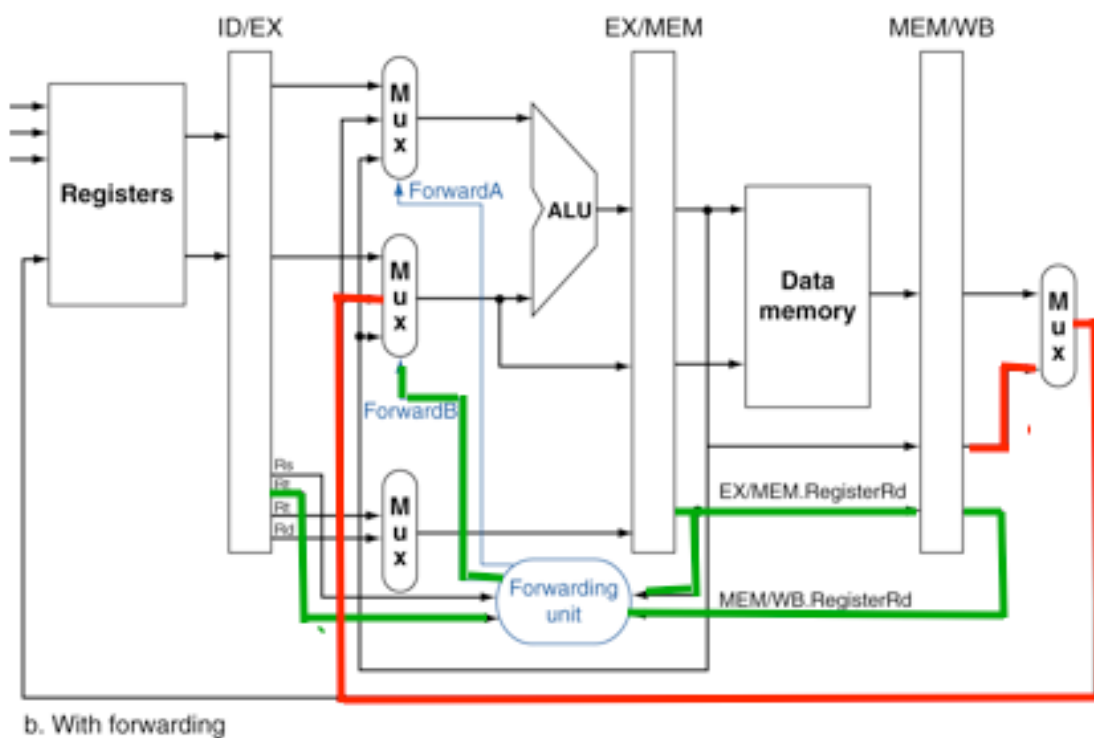
c)  $Fm_{\text{sin mejora}} = 0,25\% = 1 - Fm_1 - Fm_2 = (0,15 \times 4 + 0,1 \times 4)/4 = 0,25$

## Problemas adicionales Tema1 y Tema 2 - 2017

**P2.** En la unidad que detecta el adelantamiento (Forwarding Unit) del procesador descrito durante el curso, uno de los posibles casos a considerar está implementado con el siguiente pseudocódigo:

```
if (MEM/WB.RegWrite and MEM/WB.RegisterRd  $\neq$  0
    and EX/MEM.RegisterRd  $\neq$  ID/EX.RegisterRt
    and MEM/WB.RegisterRd = ID/EX.RegisterRt)
  then ForwardB = 01      /* si cumple condición se elige la segunda entrada del multiplexor */
  else ForwardB = 00      /* en caso contrario no se adelanta */
```

Al ejecutar una secuencia de instrucciones se ha producido el adelantamiento correspondiente al pseudocódigo anterior y se han activado las líneas marcadas en el esquema de la figura:



a) Indique una secuencia de código que se corresponda con este adelantamiento.

b) ¿Qué finalidad tiene la condición ... **and** EX/MEM.RegisterRd  $\neq$  ID/EX.RegisterRt **and** ... para el correcto funcionamiento del adelantamiento?

**Solución:**

**a)**

I1: ADD r1,r2,r3

I2: SUB r4,r0,r5

I3: XOR r6,r2,r1

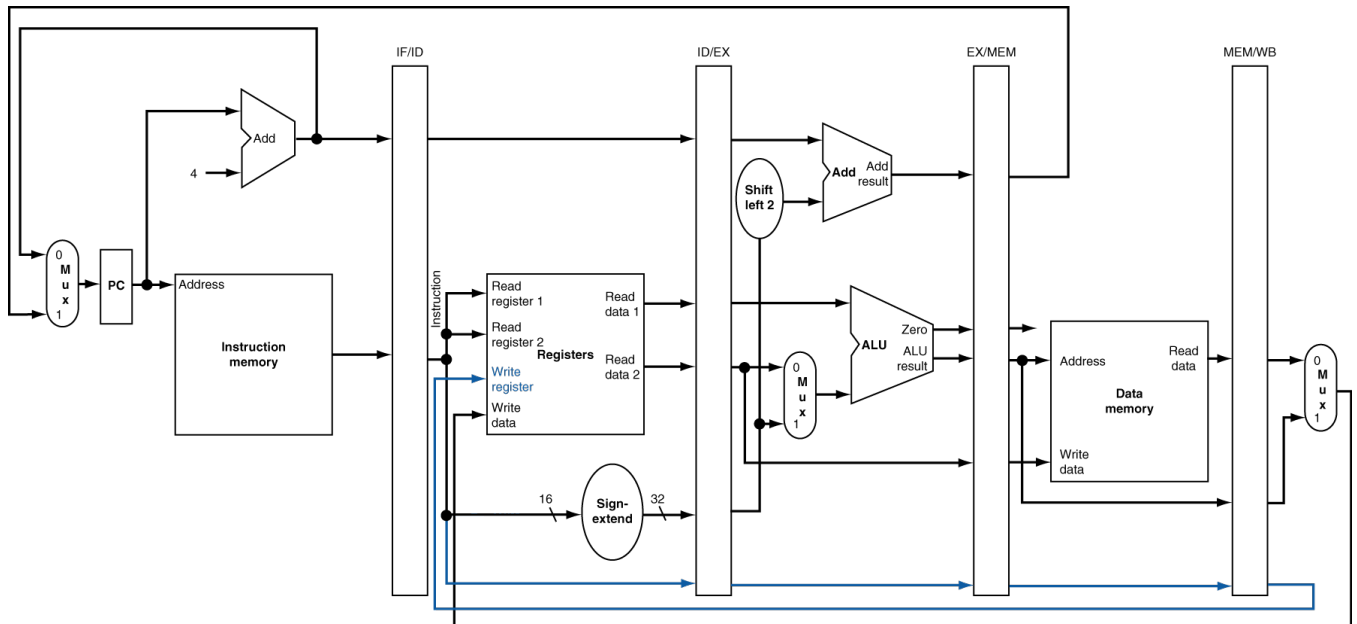
**b)**

Evitar adelantar desde MEM si hay que hacerlo desde EX.

Se activa AnticiparB (líneas verdes) para anticipa el resultado de I1 porque el fuente (ID/Ex Rt) de la instrucción (I3) en proceso (etapa EX) coincide con el destino (MEM/WB Rd) de la instrucción previa a la anterior (I1) y se adelanta el resultado guardado en los regs MEM/WB (lineas rojas). Salvo que la instrucción anterior (I2) tenga el mismo registro destino (por ejemplo si I2 fuese SUB r1,r0,r5).

## Problemas adicionales Tema1 y Tema 2 - 2017

**P3.-** La siguiente secuencia de código se ejecuta en un sistema con arquitectura Harvard y un procesador segmentado con 5 etapas como el descrito en el libro de Patterson y Hennessy: **F** (captura de instrucciones), **D** (decodificación, detección de riesgos y lectura de registros), **E** (opera en la ALU, calcula la dirección efectiva, evalúa la condición en saltos), **M** (acceso a memoria de datos y actualización en saltos del PC) y **W** (escritura en el banco de registros). El banco de registros permite escritura y lectura en el mismo ciclo.



Analice el funcionamiento del siguiente programa, empezando en la captura de la instrucción I1 y suponiendo que el salto (instrucción I3) es efectivo (salta al destino).

38	I1: SUB R10, R4, R5	; R10 = R4 - R5
3C	I2: LOAD R2, 100(R10)	; R2 = Mem(R10+100)
40	I3: BEQ R3, R2, 1	; Salta a destino si R3=R2
44	I4: LOAD R2, 104(R10)	; R2 = Mem(R10+104)
48	I5: ADDI R2, R2, 1	; R2= R2 + 1
4C	I6: AND R1, R10, R2	; R1 = R10 & R2

a) Enumere todos los posibles riesgos que se presentan en la secuencia de código y que hay que detectar al ejecutarlo en este procesador.

Riesgo de control por el salto en I3

Riesgos de datos:

RAW: I2 con I1 por R10

RAW: I3 con I2 por R2

RAW: I4 con I1 por R10

RAW: I5 con I2, I4 por R2

RAW: I6 con I1 por R10, I6 con I2, I4, I5 por R2

## Problemas adicionales Tema1 y Tema 2 - 2017

b) ¿Cuál es el destino del salto?

El destino de salto es la instrucción I5 (dirección de memoria  $48 = 40 + 4 + 1 \cdot 4$ )

Complete los siguientes cronogramas de ejecución bajo los siguientes supuestos:

c) Sin adelantamientos.

Instrucción \ tiempo	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
I1: SUB R10, R4, R5	F	D	E	M	W															
I2: LOAD R2, 100(R10)		F	D	D	D	E	M	W												
I3: BEQ R3, R2, 1			F	F	F	D	D	D	E	M	W									
I4: LOAD R2, 104(R10)						F	-													
I5: ADDI R2, R2, 1											F	D	E	M	W					
I6: AND R1, R10, R2												F	D	D	D	E	M	W		
...																				

d) Responda las siguientes preguntas:

- ¿Cuántos ciclos tarda en ejecutarse el código? ¿Cuántos ciclos de penalización se han producido?

18ciclos

ideal =  $5 + (5 - 1) = 9$  ciclos

9 ciclos de penalización

- ¿Indique concretamente que riesgos de datos han generado penalización y cuántos ciclos de detención han causado?

RAW: I2 con I1 por R10 (2 ciclos)

RAW: I3 con I2 por R2 (2 ciclos)

RAW: I6 con I5 por R1 (2 ciclos)

- ¿Indique concretamente que riesgos de control han generado penalización y cuántos ciclos de detención han causado?

Riesgo de control por el salto efectivo (3 ciclos de penalización)

Problemas adicionales Tema1 y Tema 2 - 2017
---

e) Con adelantos

[illegible]

Se decide cambiar el funcionamiento de los saltos, incorporando un comparador y un sumador para calcular el destino de salto. Con ello se puede realizar en la etapa de decodificación la evaluación del salto y cuando proceda la actualización del PC al destino.

f) Saltos evaluados en decodificación, con adelantamientos e incorporando un sistema de predicción basado en un BTB y suponiendo que acierta en la predicción de salto.

[illegible]

<b>Problemas adicionales Tema1 y Tema 2 - 2017</b>
--

**P4.-** Al ejecutar un programa de 5 millones de instrucciones, en un procesador que tiene un tiempo de ciclo de 200ns, se obtiene la siguiente distribución de instrucciones.

	Unidad INT	Unidad CF	Memoria LD/ST	Resto
%Instrucciones	50	25	15	10
CPI	2	10	2	2

Se aplica una mejora en la unidad de coma flotante (CF). Al medir el tiempo de ejecución se obtiene que la aceleración global que se consigue es de 2.

d) ¿Qué CPI se ha conseguido para las instrucciones de CF?

e) Aplicando la ley de Amdahl ¿Qué valores corresponde a los parámetros  $F_{m1}$ ,  $A_{m1}$  asociados a la mejora en CF?

f) ¿Cuál sería la máxima aceleración alcanzable mejorando sólo la unidad CF?



<b>Problemas adicionales Tema1 y Tema 2 - 2017</b>
--

a)  $A_g = 400/200 = 2$

$$N I x T_c \times (50x2 + 25x10 + 15x2 + 10x2) = 100 + 250 + 50 = 400 \times N I \times T_c$$

$$N I x T_c \times (50x2 + 25x C_{PI} + 15x2 + 10x2) = 100 + 25x C_{PI_{CF\text{mejorado}}} + 50 = 200 \times N I \times T_c$$

Para ello  $25x C_{PI_{CF}} = 50$

$$C_{PI_{CF\text{mejorado}}} = 2$$

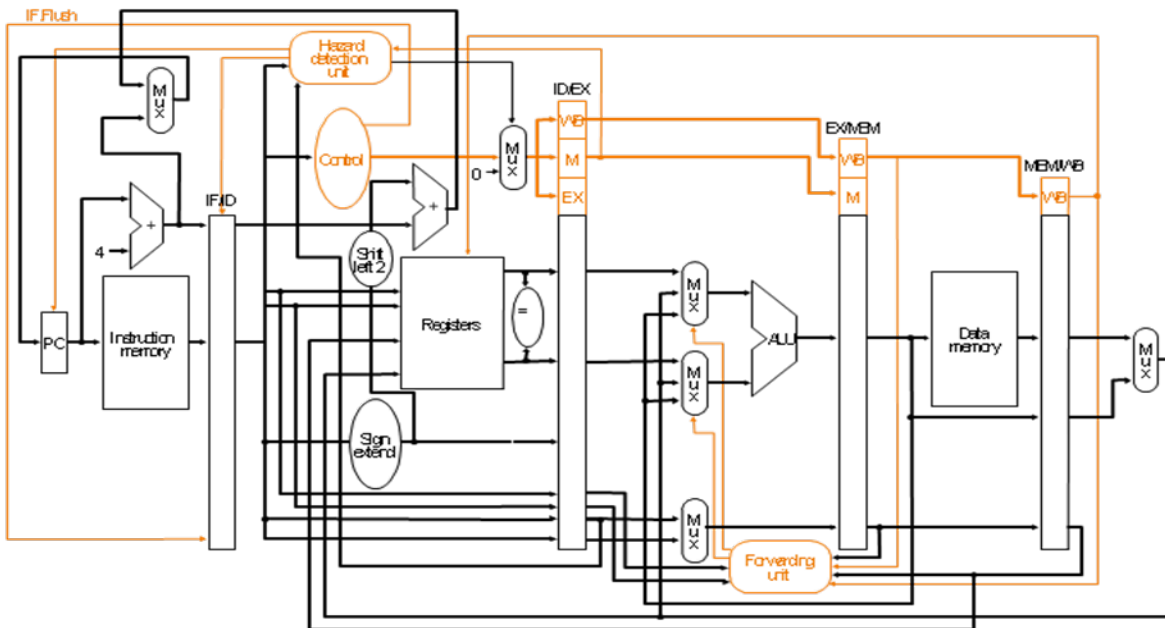
b)  $A_{m1} = 10/2 = 5;$

$$F_{m1} = (N I x T_c \times (0,25x10)) / (N I x T_c \times (0,50x2 + 0,25x10 + 0,15x2 + 0,10x2)) = 2,5/4 = 0,625$$

c) Si  $A_{m1} = \text{infinito}$  ;  $A = 1 / (1 - F_{m1}) = 1/0,375 = 2,66 = 4/1,5$

## Problemas adicionales Tema1 y Tema 2 - 2017

**P5.-** El microprocesador MIPS estudiado en teoría y realizado en las prácticas de la asignatura posee una unidad de adelantamiento de datos (data forwarding) que permite adelantar los datos de las etapas MEM y WB a la etapa de ejecución (EX):



NOTA: Los registros de las diferentes etapas se llaman PC, IF/ID, ID/EX, EX/MEM y MEM/WB respectivamente. La lógica implementada en la unidad de adelantamientos es:

<b>Adelantamientos desde la etapa MEM</b>	<b>Adelantamientos desde la etapa WB</b>
<b>if</b> (EX/MEM.EscrReg <b>and</b> EX/MEM.Reg.Rd $\neq$ 0 <b>and</b> EX/MEM.Reg.Rd = ID/EX.Reg.Rs) <b>then</b> AnticiparA = 10 <b>else</b> AnticiparA = 00  <b>if</b> (EX/MEM.EscrReg <b>and</b> EX/MEM.Reg.Rd $\neq$ 0 <b>and</b> EX/MEM.Reg.Rd = ID/EX.Reg.Rt) <b>then</b> AnticiparB = 10 <b>else</b> AnticiparB = 00	<b>if</b> (MEM/WB.EscrReg <b>and</b> MEM/WB.Reg.Rd $\neq$ 0 <b>and</b> EX/MEM.Reg.Rd $\neq$ ID/EX.Reg.Rs <b>and</b> MEM/WB.Reg.Rd = ID/EX.Reg.Rs) <b>then</b> AnticiparA = 01 <b>else</b> AnticiparA = 00  <b>if</b> (MEM/WB.EscrReg <b>and</b> MEM/WB.Reg.Rd $\neq$ 0 <b>and</b> EX/MEM.Reg.Rd $\neq$ ID/EX.Reg.Rt <b>and</b> MEM/WB.Reg.Rd = ID/EX.Reg.Rt) <b>then</b> AnticiparB = 01 <b>else</b> AnticiparB = 00

Por otro lado, como se representa en la figura, para evitar penalizaciones en los saltos se ha cambiado el diseño inicial, introduciendo en la etapa ID, un comparador que permite evaluar la condición de saltos, un sumador para calcular el destino de salto y un multiplexor para actualizar el PC caso de que sea necesario al final de esta etapa.

## Problemas adicionales Tema1 y Tema 2 - 2017

Suponga que se ejecuta la siguiente secuencia de código:

**I1:** Add r1, r2, r3  
**I2:** Sub r5, r1, r6  
**I3:** Beq r2, r1, 10

- a) ¿Puede resolverse con los adelantamientos descritos en la tabla anterior, los riesgos de datos presentes en la secuencia? ¿En caso negativo cuantos ciclos es necesario para el procesador segmentado?

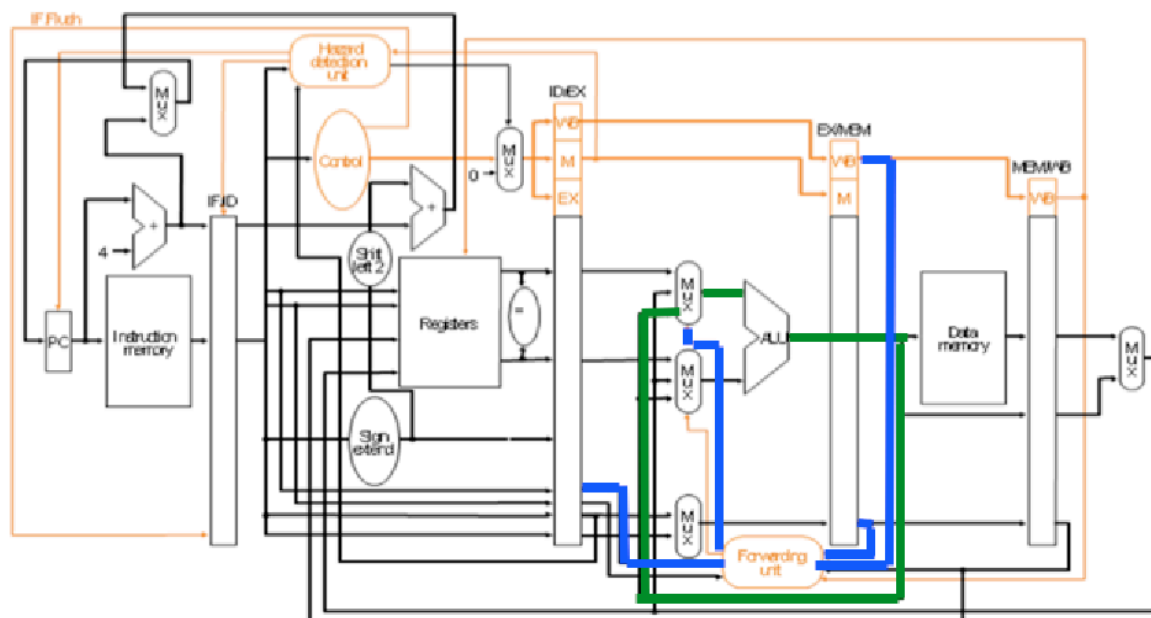
Se puede resolver el riesgo RAW entre SUB y ADD por R1 con el adelantamiento desde EX/MEM cuando EX/MEM.Reg.Rd = ID/EX.Reg.Rs = R1 y se activa la línea 01 del multiplexor anticipar A.

El riesgo de datos RAW de BEQ no se resuelve con lo implementado en la tabla y será necesario parar 1 ciclo el pipeline.

	1	2	3	4	5	6	7	8
ADD R1,R2,R3	IF	ID	EX	M	W			
SUB R5,R1,R6		IF	ID	EX	M	W		
BEQ R2,R1, 10			IF	ID	ID	EX	M	W

- b) Indique sobre la figura las líneas involucradas en el/los adelantamiento/s activados.

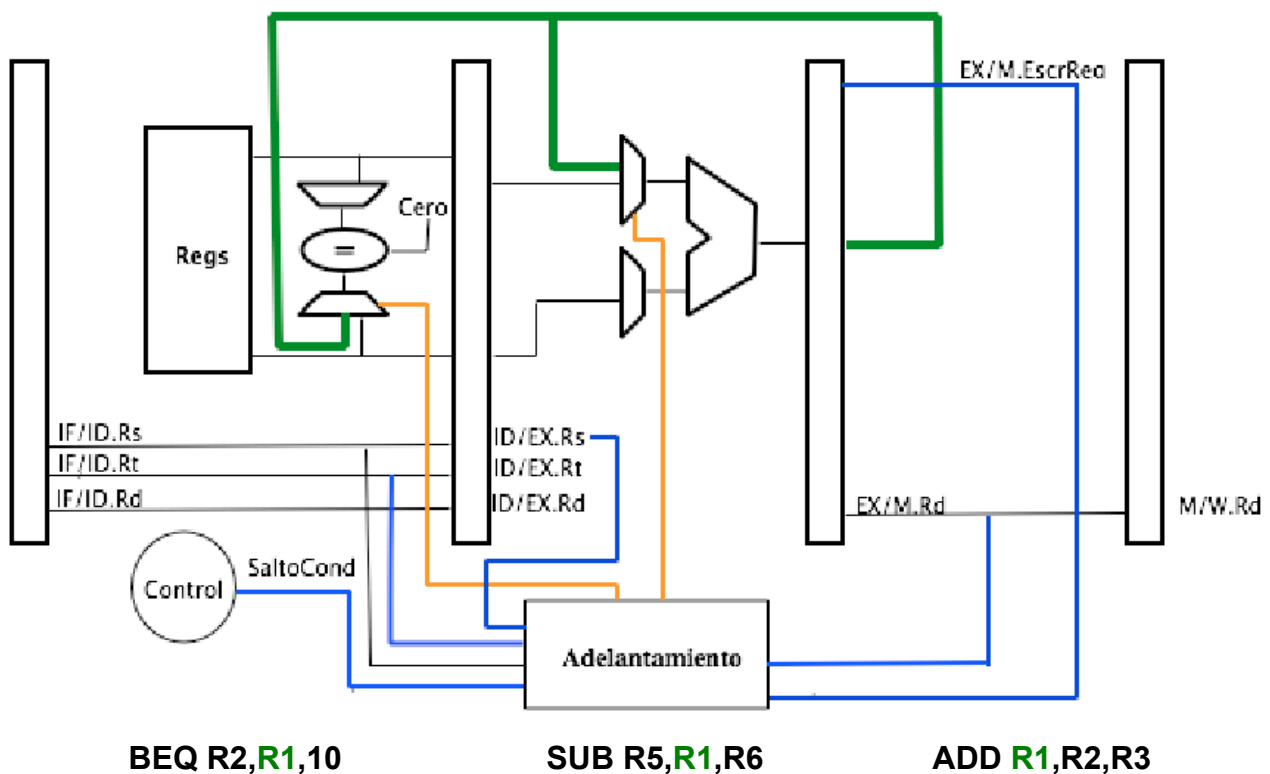
[NOTA: No es necesario espacio porque se debe responder remarcando sobre la figura.]



## Problemas adicionales Tema1 y Tema 2 - 2017

- c) Si es necesario complete la lógica de la unidad de adelantamiento para que se tenga en cuenta el riesgo de datos de la instrucción BEQ. Detalle, de una manera similar al pseudocódigo descrito en la tabla, como se detectaría el adelantamiento en este caso y represente en un esquema el camino de datos que corresponde activar.

[NOTA: Será necesario realizar una nueva figura, pero simplifique representando los elementos nuevos o modificados y las nuevas líneas de camino de datos. Aunque no se represente se supone que siguen existiendo todas las de la figura del enunciado]



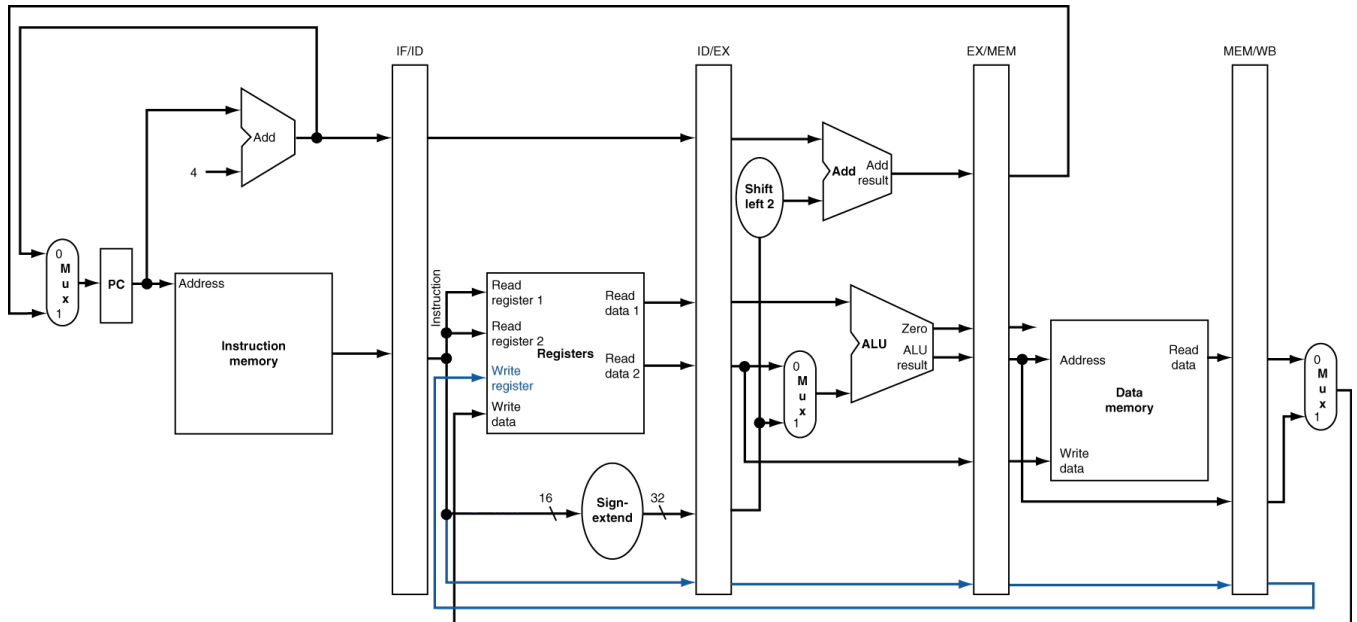
**Pseudo-Código** necesario para el adelantamiento que necesita BEQ en el ciclo 4

```

if (EX/MEM.EscrReg      and
    EX/MEM.Reg.Rd ≠ 0 and
    SaltoCond = 1      and      # Solo si es salto condicional ( detectado con señal de control en ID)
    EX/MEM.Reg.Rd = IF/ID.Reg.Rt)
then AnticiparCompB = 10
else AnticiparCompB = 00
    
```

## Problemas adicionales Tema1 y Tema 2 - 2017

**P6.-** La siguiente secuencia de código se ejecuta en un sistema con arquitectura Harvard y un procesador segmentado con 5 etapas como el descrito en el libro de Patterson y Hennessy: **F** (captura de instrucciones), **D** (decodificación, detección de riesgos y lectura de registros), **E** (opera en la ALU, calcula la dirección efectiva, evalúa la condición en saltos), **M** (acceso a memoria de datos y actualización en saltos del PC) y **W** (escritura en el banco de registros). El banco de registros permite escritura y lectura en el mismo ciclo.



Analice el funcionamiento del siguiente programa, empezando en la captura de la instrucción I1 y suponiendo que el salto (instrucción I3) es efectivo (salta al destino).

<b>38</b>	<b>I1: SUB R10, R4, R5</b>	; R10 = R4 – R5
<b>3C</b>	<b>I2: LOAD R2, 100(R10)</b>	; R2 = Mem(R10+100)
<b>40</b>	<b>I3: BEQ R3, R2, 1</b>	; Salta a destino si R3=R2
<b>44</b>	<b>I4: LOAD R2, 104(R10)</b>	; R2 = Mem(R10+104)
<b>48</b>	<b>I5: ADDI R2, R2, 1</b>	; R2= R2 + 1
<b>4C</b>	<b>I6: AND R1, R10, R2</b>	; R1 = R10 & R2

a) Enumere los posibles riesgos que se presentan en la secuencia de código y que hay que detectar al ejecutarlo en este procesador.

**Riesgo de control por el salto en I3**

**Riesgos de datos:**

**RAW: I2 con I1 por R10**

**RAW: I3 con I2 por R2**

**RAW: I4 con I1 por R10**

**RAW: I5 con I2 e I4 por R2**

**RAW: I6 con I1 por R10, I6 con I2, I4 e I5 por R2**

## Problemas adicionales Tema1 y Tema 2 - 2017

b) ¿Cuál es el destino del salto?

El destino de salto es la instrucción I5 (dirección de memoria  $48 = 40 + 4 + 1 \cdot 4$ )

Complete los siguientes cronogramas de ejecución bajo los siguientes supuestos:

c) Sin adelantamientos.

Instrucción \ tiempo	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
I1: SUB R10, R4, R5	F	D	E	M	W															
I2: LOAD R2, 100(R10)		F	D	D	D	E	M	W												
I3: BEQ R3, R2, 1			F	F	F	D	D	D	E	M	W									
I4: LOAD R2, 104(R10)						F	-													
I5: ADDI R2, R2, 1											F	D	E	M	W					
I6: AND R1, R10, R2												F	D	D	D	E	M	W		
...																				

¿Cuántos ciclos tarda en ejecutarse el código? ¿Cuántos ciclos de penalización se han producido?

18ciclos

ideal =  $5 + (5 - 1) = 9$  ciclos

9 ciclos de penalización

d) Con adelantamientos

Instrucción \ tiempo	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
I1: SUB R10, R4, R5	F	D	E	M	W															
I2: LOAD R2, 100(R10)		F	D	E	M	W														
I3: BEQ R3, R2, 1			F	D	D	E	M	W												
I4: LOAD R2, 104(R10)				F	-															
I5: ADDI R2, R2, 1								F	D	E	M	W								
I6: AND R1, R10, R2									F	D	E	M	W							
...																				

Indique los adelantamientos realizados:

Se adelanta R10 en el ciclo 4 de salida EX de I1 a entrada EX para I2.

Se adelanta R2 en el ciclo 6 de salida de etapa M de I2 a entrada de etapa EX para I3.

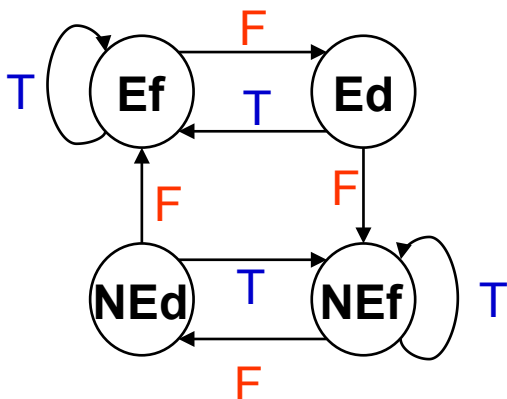
Se adelanta R2 en el ciclo 11 de salida de etapa EX de I5 a entrada de etapa EX para I6

## Problemas adicionales Tema1 y Tema 2 - 2017

e) Con adelantamientos e incorporando un sistema de predicción basado en un BTB y suponiendo que acierta en la predicción de salto.

Instrucción \ tiempo	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
I1: SUB R10, R4, R5	F	D	E	M	W															
I2: LOAD R2, 100(R10)		F	D	E	M	W														
I3: BEQ R3, R2, 1			F	D	D	E	M	W												
I4: LOAD R2, 104(R10)																				
I5: ADDI R2, R2, 1				F	F	D	E	M	W											
I6: AND R1, R10, R2						F	D	E	M	W										
...																				

f) Suponga que el BTB tiene 16 entradas inicialmente vacías. Represente en una figura los campos que forman el BTB y el contenido de los mismos después de ejecutar el salto, suponiendo que el salto ha sido efectivo, y la predicción que se realizó por acierto en el BTB fue la correspondiente al estado Ed.



Estados	Codificación	Predicción	
Efectivo fuerte (Ef )	11	Efectiva ( Salta)	
Efectivo débil( Ed)	10	Efectiva (Salta)	Estado inicial
No Efectivo débil ( NEd)	01	No Efectiva ( No salta)	
No Efectivo fuerte ( NEf)	00	No Efectiva ( No salta)	

**SOLUCION:** Estructura y contenido de los campos del BTB después del salto

Dirección de salto	Destino de salto	Predicción
40	48	11
...hay 16 entradas	...	..
...		