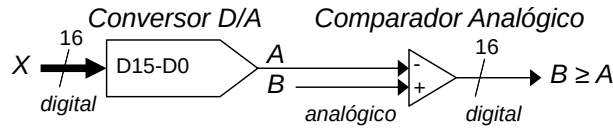


Pregunta 1



La figura muestra un conversor digital/análogo (D/A) y un comparador analógico. El conversor digital/análogo recibe como entrada un número entero binario X en D15-D0 (16 bits) y entrega en su salida una señal analógica A consistente en un voltaje entre 0 y 12V proporcional al valor X de la entrada, es decir $12V * X / 2^{16}$. El comparador recibe dos señales analógicas A y B entre 0 y 12V y las compara entregando en su salida un valor binario 1 que indica que la señal B (entrada +) tiene un voltaje superior a A (entrada -) o 0 en caso contrario. En esta pregunta Ud. medirá el voltaje de la señal B comparándolo con múltiples voltajes establecido mediante el conversor D/A.

Parte a.- Implemente una interfaz de entrada/salida de ambos conversores para LRV32IM con un solo puerto de entrada/salida en la dirección 0xc0000000. Cuando se escriba X en ese puerto, debe almacenar X en un registro de 16 bits que será la entrada del conversor D/A. Cuando se lea del mismo puerto debe obtener un bit que indica si el voltaje en B es mayor o igual a $12V * X / 2^{16}$.

Parte b.- Programe la función: `double voltaje();`

Esta función debe usar el puerto de la parte a para medir y retornar el voltaje en B usando búsqueda binaria.

Parte c.- Considere que la salida del comparador está conectada a `irq2`. Programe la función: `void alerta(double A, void (*f)());`

Esta función debe configurar la interfaz de la parte a para que se gatille una interrupción cuando el voltaje en B iguale o supere A . Programe también la rutina de atención de la interrupción de manera que se inhiba `irq2` y luego se invoque la función `f`. Necesitará usar la función:

```
void set_irq_handler(int irq_id, void (*hdlr)());
```

Si `hdlr` es NULL, se inhibe ese `irq`. Use variables globales.

Pregunta 2

Parte i.- La figura muestra un extracto del estado actual de un *caché* de 4 KB (2^{12} bytes) de 1 grado de asociatividad con 256 líneas de 16 bytes. Por ejemplo en la línea 7f del caché (en hexadecimal)

línea cache	etiqueta	contenido
25	425	
7f	97f	
b8	1b8	

se almacena la línea de memoria que tiene como etiqueta 97f (es decir, la línea que va de la dirección 97f0 a la dirección 97ff). Un programa accede a las siguientes direcciones de memoria (en hexadecimal): 1b80, 425c, 97f8, 3b80, 4250, 3b88, 6258, 1b80 y 97f0. **Indique** qué accesos a la memoria son aciertos en el caché, cuáles son desaciertos y **rehaga la figura** mostrando el estado final del caché. Por ejemplo el acceso 1b80 es un acierto.

Parte ii.- Considere el diseño de LRV32IM microprogramado que usó en sus tareas. Al reverso del enunciado está el diagrama de bloques. El registro de instrucción *Inst* tiene cargada la instrucción `addi s1, a2, t3` ($s1$ es el registro de destino). **Evalúe** de manera independiente si cada una de las 5 transferencias entre registros del cuadro de la derecha se puede realizar en *un solo ciclo del reloj*. Si la transferencia se puede realizar indique **cuáles son la señales de control** requeridas para llevarla a cabo. Si no se puede realizar, explique por qué.

- a) $s1 \leftarrow a2 + 4$
- b) $Inst \leftarrow Mem[a2+t3]$
- c) $s1 \leftarrow a2 \& t3$
- d) $a2 \leftarrow s1 + t3$
- e) $Mem[a2] \leftarrow t3$

Pregunta 3

La tabla de la derecha muestra las instrucciones Risc-V ejecutadas por un programa.

Parte a.- Haga un diagrama, como el de la figura con título *Arq. en pipeline*, que muestre el ciclo en que se ejecuta cada etapa de las instrucciones, considerando una arquitectura en pipeline con etapas *fetch*, *decode* y *execute*, y con predicción de saltos. Considere que en D sí salta y la predicción es que el salto no ocurre.

Parte b.- Repita el mismo diagrama de la parte a considerando que la predicción es que el salto sí ocurre.

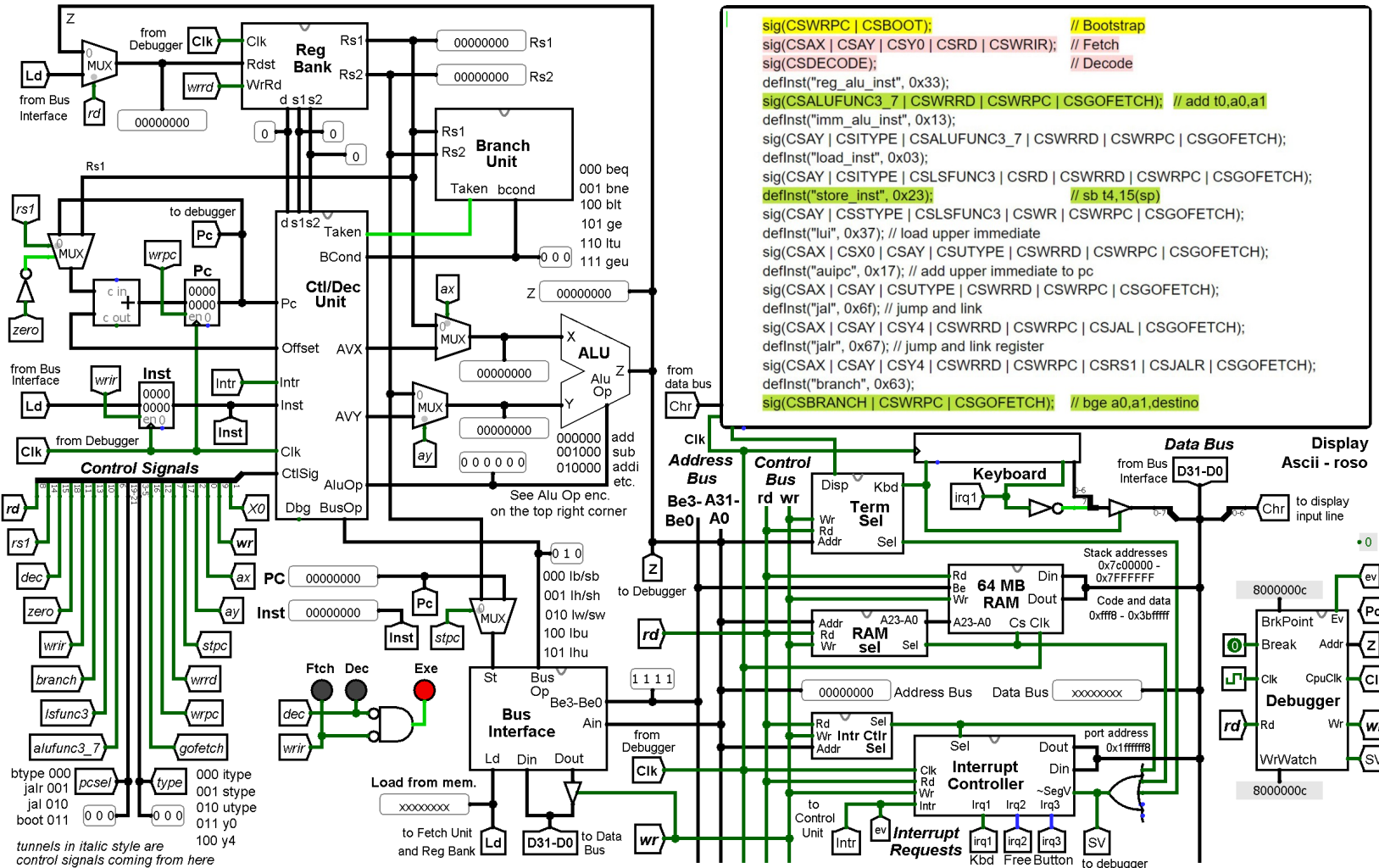
Parte c.- Haga un nuevo diagrama, como el de la figura de la derecha, considerando una arquitectura superescalar con 2 pipelines y las mismas etapas que la parte a. Considere que el salto en D ocurre y la predicción es que sí salta.

```
Programa
A and a5,s4,t0
B addi a4,t1,4
C sub a6,t2,a5
D bgt a6,a5, M
E
F
...
M sub a2, s0, a0
N ori a1, a1, 4
```

Arq. en pipeline			
	Fetch	Dec	Exe
1	A		
2	B	A	
3	C	B	A
...			

Arq. superescalar			
	Fetch	Dec	Exe
1	AB		
2	CD	AB	
...			

CC4301 Arquitectura de Computadores – Control 3 – Primavera 2024 – Prof.: Luis Mateu



Alu Op encoding
 bit 5 (left): mul/div operation
 bit 4: immediate format
 bit 3: subtract / sra
 bit 2-0 (right): func3

Alu Op func3 encoding
 000 add/sub 100 xor
 001 sll 101 srl/sra
 010 slt 110 or
 011 sltu 111 and

LRV32IM CPU
 Risc-V 32 bit integer base instruction set with multiply/divide extension
 Simple microprogrammed physical architecture

64 MB RAM
 No Memory Management Unit
 By Luis Mateu
 Previous work by leijurv
 GPL Licence

Display
 Ascii - rojo

