

CC41C Introducción al Hardware
Examen - Otoño 2009
Profesor: Luis Mateu

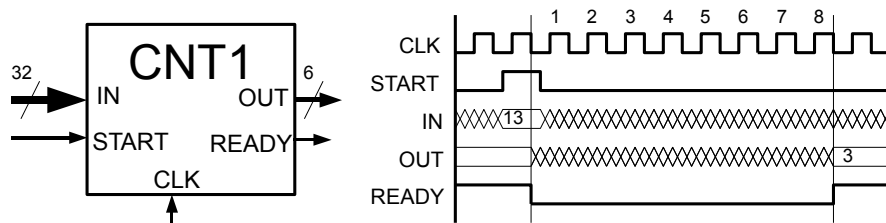
Pregunta 1

Se desea agregar la instrucción JMPL a M32:

Notación Assembler	Formato instrucción	Significado
JMPL [reg1 + imm], regd		$regd = PC$ $PC = reg1 + imm$
JMPL [reg1 + reg2], regd		$regd = PC$ $PC = reg1 + reg2$

Especifique ciclo por ciclo las señales de control que son necesarias para ejecutar JMPL (no es necesario que especifique las señales para la carga de la instrucción y la decodificación).

Pregunta 2



El circuito CNT1 de la figura recibe como entrada IN un número de 32 bits y entrega como salida OUT la cantidad de bits en 1 que aparecen en IN. CNT1 inicia el conteo cuando START se pone en 1. Una vez transcurridos 8 ciclos, CNT1 entrega el resultado y coloca la línea READY en 1. Las salidas deben permanecer constantes hasta que se solicite un nuevo conteo.

Implemente este circuito usando diseño modular. Ud. debe realizar el conteo en exactamente 8 ciclos. Menos ciclos resultaría demasiado caro y más ciclos demasiado ineficiente.

Pregunta 3

Parte a.- Considere el circuito CNT1 de la pregunta 2 como un dispositivo de E/S. Implemente una interfaz mapeada en memoria de CNT1 para un computador con buses de datos y direcciones de 32 bits. Elija Ud. la dirección del puerto de E/S de CNT1. Su interfaz debe permitir que un programa suministre un dato para IN, active la señal START (basta hacerlo por el tiempo que se activa la línea WR de la CPU) y obtenga los valores OUT y READY.

Parte b.- Escriba en C la función `int contar(int val)` que utiliza la interfaz de la parte a.- para calcular eficientemente la cantidad de 1s en `val`. No olvide realizar un ciclo de busy-waiting para esperar a que READY se coloque en 1.

Pregunta 4

i. La figura muestra a la izquierda un programa que se ejecuta en un procesador Risc, con una implementación superescalar de grado 2 y 5 etapas en sus 2 pipelines: *fetch*, *decode*, *analyze*, *execute* y *store*. Acceder a la memoria toma 2 ciclos del reloj.

```

loop:
  shiftl R1, 2, R2
  load [R6+R2], R3
  add R1, 1, R1
  add R3, R4, R4
  cmp R1, R5
  blt loop
  ...
  
```

instrucción	ciclo				
	1	2	3	4	5
shiftl R1,2,R2	F	D	A	...	
load [R6+R2],R3	F	D	A		
add R1,1,R1		F	D		
add R3,R4,R4			F	D	
.			.		
.			.		
.			.		

Complete el diagrama de ejecución de la derecha, considerando 2 iteraciones del ciclo `loop`.

ii. Indique en su diagrama (si es aplicable) en que momentos se usa (a) *register bypassing*, (b) *register scoreboarding*, (c) *register renaming*, (d) *predicción de saltos*, y (e) *ejecución especulativa*.