

Auxiliar 7

Profesor: Pablo Guerrero.
Auxiliar: Ian Yon
Viernes 10 de julio de 2015



Problema 1: instrucciones M32

A continuación se indican 8 conjuntos de transferencias entre registros para el diseño actual de M32. Considerando cada uno de estos conjuntos en forma independiente, señale cuáles de ellos se pueden llevar a cabo en un solo ciclo del reloj y cuáles no. Si un conjunto de transferencias es válido indique las señales de control necesarias para realizarlas. Cuando un conjunto no es válido explique porque (no necesita dar las señales de control).

1. AR = PC ; PC = PC+4
2. PC = 4 ; AR = 4
3. R1 = PC cuando IR contiene la instrucción ADD R2, R3, R1 (R1= R2+R3)
4. PC = R1 + R3 cuando IR contiene ADD R2, R3, R1
5. R1 = PC + R2 cuando IR contiene ADD R2, R3, R1
6. IR = R2 + R3 cuando IR contiene ADD R2, R3, R1
7. PC = Mem[AR]
8. R1 = R2 + R3; IR= Mem[AR] cuando IR contiene ADD R2, R3, R1

Problema 2: extensión M32

Se desea agregar la instrucción LDRPC a M32 (load relative to pc). Esta instrucción lee una palabra de 32 bits en la memoria suministrando un desplazamiento relativo al registro PC, es decir:

| Assembler | Instrucción | Formato instrucción | | |
|---------------------------------|---|---------------------|-------|------|
| | | 31-24 | 23-19 | 18-0 |
| LDRPC <i>imm, r_i</i> | $r_i \leftarrow \text{Mem}[\text{PC} + \text{imm}]$ | Instrucción | r_i | imm |

No es posible implementar LDRPC con el diseño físico actual, puesto que no se puede llevar el desplazamiento de 19 bits a la ALU.

1. Modifique e implemente el módulo Y-SEL para que esta instrucción sea implementable. Su modificación debe ser tal que las actuales instrucciones de M32 continúen siendo implementables.
2. Indique ciclo por ciclo las transferencias entre registros y las señales de control necesarias para ejecutar LDRPC. No incluya la fase de fetch ni la fase de decodificación.

Problema 3: Control 3 2011-1

En la figura 1 se muestra un circuito guardián que se encarga de vigilar que la temperatura ambiente no salga de un rango dado. El guardián tiene dos puertas de E/S que permiten

especificar la mínima y la máxima temperatura aceptable. Cuando la temperatura se sale de estos parámetros, el sensor activa la línea INT.

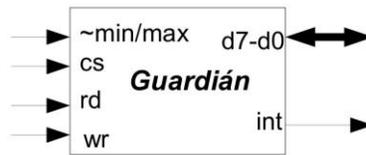


Figura 1: Circuito guardián

El siguiente procedimiento sirve para configurar el guardián:

```
void setTempRange(char min, char max){
    /*min y max son enteros de un byte*/
    char *port_min = (char*)0xFF00;
    char *port_max = (char*)0xFF01;
    *port_min = min;
    *port_max = max;
}
```

Los valores 0 y 255 se usan para indicar al guardián que no debe producir ninguna interrupción.

1. Implemente la interfaz que se necesita para conectar el guardián con el bus del procesador, de modo que el procedimiento anterior funcione correctamente. Considere un microcontrolador con un bus de datos de 8 bits y un bus de direcciones de 16 bits.
2. Implemente en C la rutina de atención `alertTemp()` que debe procesar la interrupción causada por el guardián. Esta rutina debe desplegar un mensaje en pantalla (usando `printf`) y evitar que se siga produciendo la interrupción.
3. (Propuesta) Considere que cuando se lee en las posiciones de timer se lee el valor de la máxima y mínima temperatura configuradas en el guardián. Ahora, implemente la función pedida en el punto anterior, pero después de imprimir, que se restauren las interrupciones y deje al guardián con la misma configuración que antes de la interrupción.