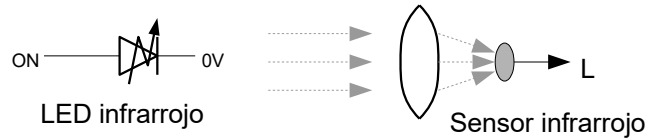


Pregunta 1

Un sensor de luz infrarroja detecta el bloqueo de un haz de luz emitido por un LED infrarrojo.



Este sensor se usa en sistemas de alarmas para detectar la presencia de intrusos. Cada alarma usa un LED infrarrojo que tiene una entrada ON y un sensor de luz infrarroja que tiene una salida L. Cuando ON está en 1, el LED emite un haz de luz dirigido hacia el sensor. El LED se apaga si ON está en 0. L es 1 cuando el sensor detecta el haz de luz, 0 en caso contrario. La presencia de un intruso está caracterizada cuando ON es 1 y L es 0. Un intruso no detectará la presencia de la alarma porque la luz infrarroja no es visible para los humanos.

Parte a.- Considere un sistema de alarmas con 8 LEDs y 8 sensores para detectar intrusos en 8 zonas de una habitación. Implemente una interfaz de entrada/salida que permita encender o apagar los 8 LEDs y que permita determinar el estado de los 8 sensores. Ud. define la o las direcciones del o los puertos de entrada/salida que requiera.

Parte b.- Use la parte a para programar las funciones `void confLEDs(int leds)` e `int leerSensores()`. La función `confLEDs` recibe en `leds` una máscara de 8 bits. Cada bit se asocia con uno de los LEDs. Un bit 1 enciende ese LED, un 0 lo apaga. La función `leerSensores` retorna una máscara de 8 bits. Cada bit se asocia con uno de los sensores. Un bit en 1 indica que el sensor detecta el haz de luz.

Parte c.- Modifique la interfaz de la parte a para que se gatille una interrupción cuando un sensor no detecta la luz emitida por alguno de los LEDs *que están encendidos*. Use la entrada `irq2` del controlador de interrupciones.

Parte d.- Programe la función: `void alerta(int leds, void (*f)());`

Esta función debe configurar una interrupción para que se invoque `f()` cuando se detecte la presencia de algún intruso que bloquea alguno de los LEDs especificados en la máscara `leds`.

Pregunta 2

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 a3, t4, 27` (`a3` 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 las señales de control** requeridas para llevarla a cabo. Si no se puede realizar, explique por qué.

- a) $t4 \leftarrow a3 + 27$
- b) $a3 \leftarrow t4 + 27$
- c) $Inst \leftarrow t4 + 27$
- d) $a3 \leftarrow Mem[t4 + 27]$
- e) $a3 \leftarrow pc + 27$

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 el salto en E no ocurre, que el salto en F sí ocurre y que la predicción es exitosa en ambos casos.

Parte b.- Repita el mismo diagrama de la parte a considerando que la predicción es exitosa para el salto en E, pero falla para F.

Parte c.- Haga un nuevo diagrama, como el de la figura con título *Arq. superescalar*, considerando una arquitectura superescalar con 2 pipelines y las mismas etapas que la parte a. Considere que el salto en E no ocurre, que el salto en F sí ocurre y que la predicción es exitosa en ambos casos.

Programa			
A	add	a1, a2, a3	
B	andi	t4, a1, 7	
C	addi	t5, t4, 1	
D	add	t6, t4, a4	
E	blt	a1, t4, I	
F	beq	t5, a1, P	
G	add	...	
H	sub	...	
I	xor	...	
J	andi	...	
...			
P	addi	a3, a2, 1	
Q	sub	t7, a4, t5	

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 2025 – Prof.: Luis Mateu

