

Taller 110,1

Preparación del Primer Parcial

Pregunta 1

Indique dos diferencias entre CPUs construidas por lógica cableada y microprogramada. ¿Cuál implementación se usa más comúnmente en RISC? ¿Y en CISC? Justifique

Pregunta 2

Utilizando el algoritmo de suma de punto flotante, indique la representación en punto flotante Media Precisión del resultado de sumar 10,25 y el siguiente número cuyo código binario en la misma representación es

1101000011010000.

Pregunta 3

Defina el formato de instrucción y los códigos de operación para la siguiente arquitectura RISC de 16 bits con 8 registros.

Instrucción	Resultado	Instrucción	Resultado	Instrucción	Resultado
AND RegX, RegY, RegZ	RegZ = RegX AND RegY	ADD RegX, RegY, RegZ	RegZ = RegX + RegY	LOAD RegX, RegY	RegY = Mem[RegX]
OR RegX, RegY, RegZ	RegZ = RegX OR RegY	SUB RegX, RegY, RegZ	RegZ = RegX - RegY	STORE RegX, RegY	Mem[RegY] = RegX
NOT RegX, RegY	RegY = NOT RegX	DIV RegX, RegY, RegZ	RegZ = RegX DIV RegY	MOV RegX, RegY	RegY = RegX
XOR RegX, RegY, RegZ	RegZ = RegX XOR RegY	MOD RegX, RegY, RegZ	RegZ = RegX MOD RegY	MOVI Inm8, RegX	RegX = Inm8(8 Bits)

Pregunta 4

Dibuje un flip-flop RS y justifique mediante ejemplos por qué es un circuito secuencial.

Problema 1

Se pretende almacenar en una memoria ROM la parte entera del valor medio de dos números enteros representados en binario de 8 bits.

a) Especificar tamaño de la ROM, entradas/salidas y construirla disponiendo de compuertas básicas y de las siguientes memorias:

- 1 ROM 32Kx8
- 1 ROM 32Kx4
- 2 ROMs 32K x 2

b) Implementar en alto nivel el código de carga de la ROM

Problema 2

La paridad vertical y horizontal es un método para detectar errores, el cual permite corregir errores de hasta un bit. Dada una matriz de bits, se calcula la paridad de cada fila y cada columna. Para este ejercicio se considerarán solo matrices de 8x8. Por ejemplo, dada la siguiente matriz, se presentan a la derecha la paridad horizontal y abajo la paridad vertical.

```
11010101 1
00110101 0
10111110 1
00110101 0
10110100 0
10001110 0
10110101 1
00110110 0
```

```
11010010
```

Escriba una función que dada una matriz de 8x8 bits, representada como un arreglo de caracteres, su paridad horizontal y su paridad vertical, indique si hay algún error en la matriz o no y dónde. La respuesta se codifica en 8 bits, donde el bit más significativo indique si hubo o no error en la transmisión, los bits [2..0] indiquen la columna del error y los bits [5..3] indiquen la fila del error.

```
char hayError(char matriz[8], char horizontal, char vertical);
```