Compendio de respuestas primer parcial

Arquitectura de Computadoras

2022

Recordar que:

- **El parcial consta de 5 preguntas.**
- **Para aprobar el curso debe contestar correctamente una pregunta en cada parcial.**
- **Para obtener la exoneración parcial debe contestar correctamente tres preguntas en cada parcial.**

Pregunta 1

A) La representación media precisión utiliza el bit más significativo para el signo, luego cinco bits para el exponente y los diez bits menos significativos para la mantisa (signo = 1, exponente = 10010 (d=15) y mantisa = 0011000000). Por lo tanto el resultado numérico es:

-1,0011 . 2^3b = -1001,1b = -9,5.

B) Se representa en complemento a dos tomando los ocho bits menos significativos como la parte fraccionaria. Al descomplementar la representación se obtiene el código 0011011101000000. Tomando sus ocho bits menos significativos como la parte fraccionaria se obtiene el resultado numérico es:

-110111,01b = -55,25.

C) El código representa un número positivo. Para obtener el resultado numérico debe restarsele el desplazamiento (d=2^{15} = 1000000000000000). Por lo tanto el resultado numérico es:

100100011000000b = 18624.

Pregunta 2

1. $125{,}8125_{(10)} = 1111101{,}1101_{(2)}$, multiplicando por 2^7, expresando en 16 bits (como es positivo no se precisa hacer el complemento a 2), obtenemos:

0011111011101000

2. En precisión simple usamos 1 bit para el signo, 8 para el exponente y 23 para la mantisa. Para normalizar debemos multiplicar por 2^6 (1,1111011101 x 2^6) con lo que la representación queda:

0 10000101 11110111010000000000000.

Pregunta 3

Considerando el ordenamiento de bits que forman cada paquete de Hamming de 4 bits a4a3a2p3a1p2p1 podemos calcular el síndromes de cada paquete (x es xor):

s2=p3xa4xa3xa2
s1=p2xa4xa3xa1
s0=p1xa4xa2xa1

* 0x12 = 0010010 paquete: 0010

s2=0x0x0x1 = 1
s1=1x0x0x0 = 1
s0=0x0x1x0 = 1

Por lo tanto el bit 7 está mal y se corrige a 1010010 = 0x52 => paquete 1010

* 0x3D = 0111101 paquete: 0111

s2=1x0x1x1 = 1
s1=0x0x1x1 = 0
s0=1x0x1x1 = 1

Por lo tanto el bit 5 está mal y se corrige a 0101101 = 0x2D => paquete 0101

* 0x52 = 1010010 paquete 1010.

No tiene error (el síndrome dará 0 pues corresponde al primer paquete ya corregido)

Pregunta 4

a) La distancia entre dos representaciones binarias se define como el número de bits distintos entre los dos códigos.
La distancia entre las dos representaciones binarias es 2. Los bits número cuatro y diez son los únicos diferentes.

b) La cantidad de bits 'p' de redundancia requeridos, de modo de tener la capacidad de corrección de errores, para un código de tamaño 'k' está dado por la siguiente ecuación:

$$2^p >= p + k + 1$$

El razonamiento detrás de esta afirmación es que para poder identificar el bit que tiene error dispondremos de 2^p

combinaciones posibles, generadas por los p bits. Con esas combinaciones debemos poder señalar cual es el bit errado de los p + k bits que tendrá el código completo (el original más los bits de redundancia agregados). Además precisaremos tener una combinación para indicar que no hay error y de ahí surge que el lado derecho de la expresión es p + k + 1.

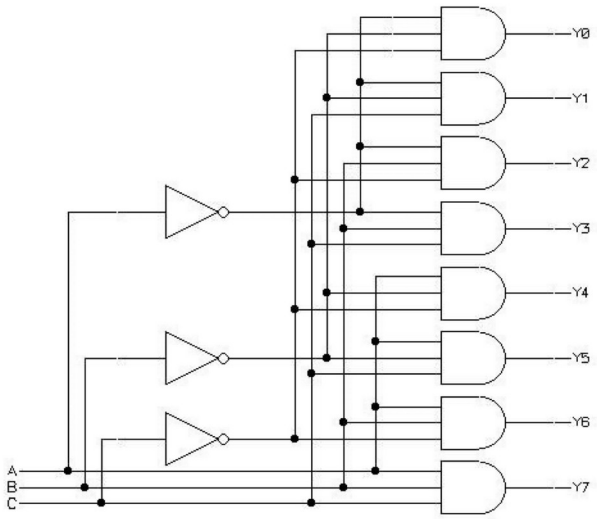
En el caso de un código de 32 bits, la cantidad mínima de bits de redundancia para que se cumpla la condición sería 6, ya que:

$2^6 >= 6 + 32 + 1$

Por lo tanto 6 es la cantidad de bits de redundancia mínima para este caso.

Pregunta 5

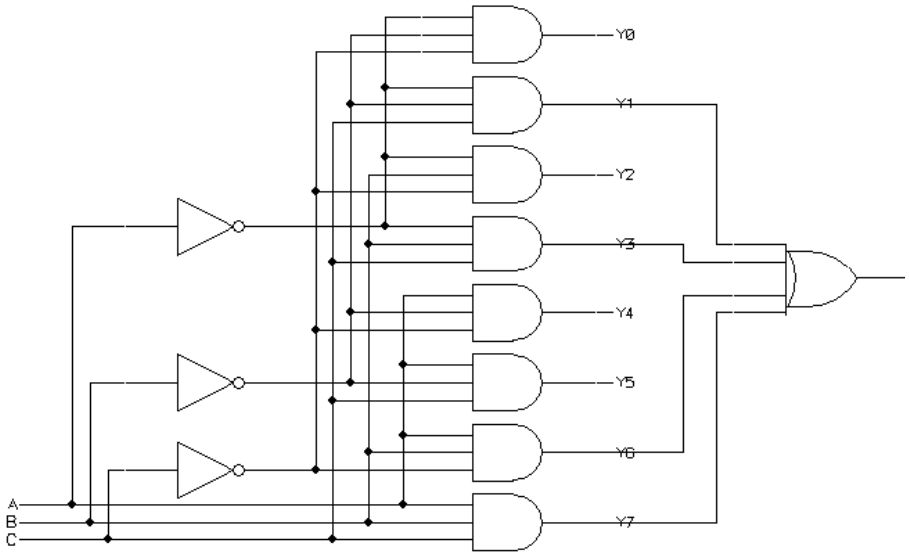
a) El circuito de un decodificar de 3 bits es el siguiente:



b) La función f tiene la siguiente tabla de verdad

a	b	c	ab + !ac
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Por lo que el circuito se puede construir en base al decodificador y una compuerta OR de 4 entradas:



Nota: no es necesario repetir el circuito interno del decodificador, es válido reemplazarlo con una caja con sus correspondientes entradas y salidas.

Pregunta 6

a) Los diagramas de Karnaugh se utilizan para minimizar expresiones lógicas mediante un método sistemático. En la construcción de circuitos es una herramienta utilizada, ya que al modelar circuitos con funciones lógicas es posible minimizarlas para construir un circuito equivalente (que ante las mismas entradas presente las mismas salidas), lo cual permite ahorrar compuertas para su construcción.

b)

f(a,b,c,d) = σ(0,1,2,4,5,6,8,10,11,12,14,15)

a	b	c	d	f
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

ab\cd	00	01	11	10
00	1	1	0	1
01	1	1	0	1
11	1	0	1	1
10	1	0	1	1

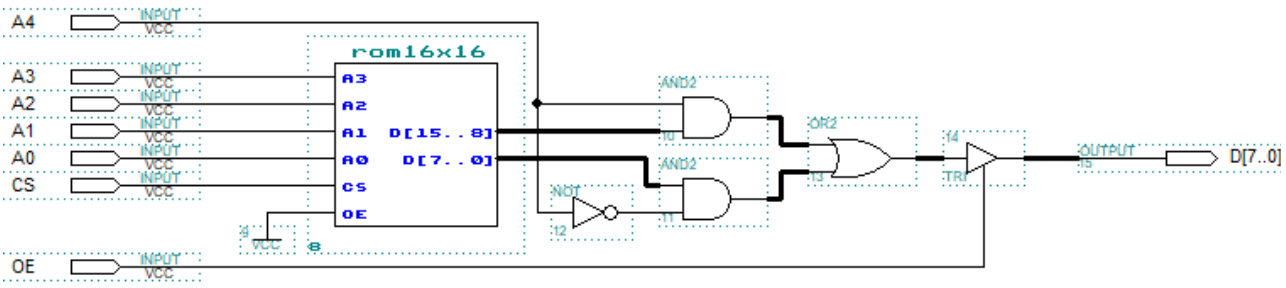
F = $\overline{a}\,\overline{c} + ac + \overline{d}$

Pregunta 7

a)
Output Enable (OE): la entrada Output enable tiene como cometido habilitar o no la salidas de una ROM, desde el punto de vista eléctrico. Opera de la siguiente manera: cuando dicha entrada de control está en 0, la salida pasa al "estado de alta impedancia" o “tercer estado” (no influye en el circuito) y cuando la entrada de control está en 1, la salida está en estado lógico 0 ó 1. Esto nos permite realizar ORs “cableados” (sin necesidad de utilizar compuertas).

Chip Select (CS): cuando OE=1, si CS = 0 todas las salidas de la ROM están en 0, con independencia de las entradas de dirección y del "valor" almacenado en la "posición" de la ROM indicada por dicha dirección y si CS = 1 las salidas presentan el contenido de la ROM en la posición señalada por la dirección. Esto nos permite ahorrar en la implementación la estructura de ANDs que deberíamos colocar a la salida de las ROMs a la hora de trabajar con varias ROMs y elegir la salida de la ROM general.

b)

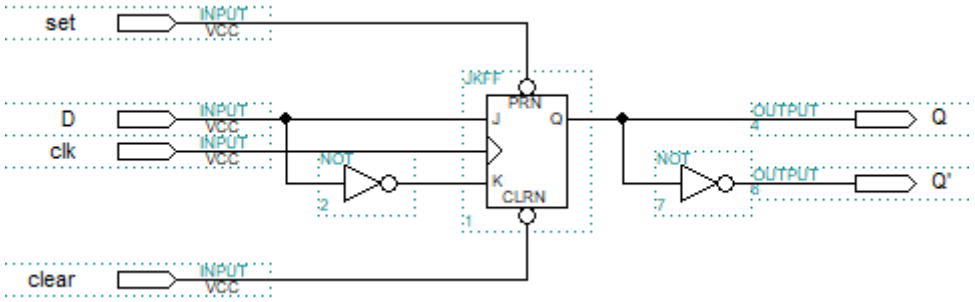


Pregunta 8

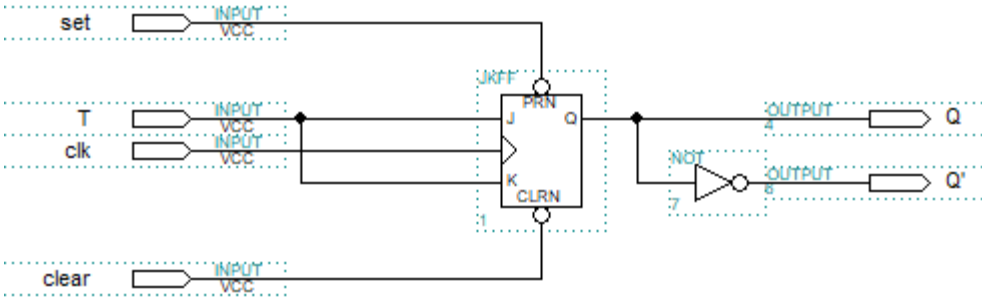
La tabla de la verdad reducida del flip flop JK es la siguiente:

J	K	Q(n+1)
0	0	Q(n)
0	1	0
1	0	1
1	1	Q(n)’

A partir de ella armamos los circuitos pedidos:
a) Colocamos J=D y K=D' y logramos cumplir la ecuación del FF-D: Qn+1 = Dn



b) Colocamos J=K=T y logramos cumplir la ecuación del FF-T: Qn+1 = Tn'Qn + TnQ'n



Pregunta 9

La CPU es la unidad central de procesamiento, la encargada de ejecutar las instrucciones en una máquina de Von Neumann.
La misma contiene:

- ALU o unidad aritmético-lógica, circuito combinatorio encargado de realizar diferentes operaciones.
- Banco de Registros, elementos de memoria de rápido acceso.
- Unidad de Control, circuito secuencial encargado de llevar adelante el ciclo de instrucción.

La CPU es un circuito SECUENCIAL dado que sus salidas no solo dependen de las entradas actuales, sino también de las entradas anteriores. Un ejemplo claro de esto, es que lo que una computadora hace (salidas de la CPU) depende de varias instrucciones leídas de memoria (entradas anteriores a la CPU).

Pregunta 10

El modelo de memoria de 8086 es "segmentado". Este término significa que en la memoria se identifican segmentos de 64 KBytes de tamaño asociados a registros especiales de 16 bits llamados registros de segmento, y

solo se puede, en cada momento, direccionar posiciones de memoria dentro de cada segmento. Los registros de segmento son el CS (code segment), DS (data segment), SS (stack segment) y ES (extra segment).

Las direcciones físicas se forman a partir de un registro de segmento y un desplazamiento (también de 16 bits, por ser el segmento de 64 KBytes), con la siguiente operación:

dirección_física = segmento * 16 + *desplazamiento*.

Esto da como resultado una dirección de 20 bits.