Examen de Arquitectura de Computadoras

16 de febrero de 2017

Instrucciones:

- Indique su nombre, apellido y número de cédula en todas las hojas que entregue.
- Escriba las hojas de un solo lado.
- Las hojas deben estar numeradas y en la primer hoja debe escribirse el total.
- Empiece cada ejercicio en una hoja nueva.
- No puede utilizar material ni calculadora.
- **Apague su celular.**
- La duración del examen es de tres horas. En dicho tiempo debe también completar sus datos.
  Solo se contestarán dudas de letra. No se aceptan preguntas en los últimos 30 minutos del examen.
- Para aprobar debe contestar correctamente un ejercicio entero y dos preguntas.

Pregunta 1

Expresar la función mayoría de tres variables en notación Σ (sigma = suma de productos canónicos) y minimizarla utilizando Karnaugh.

Pregunta 2

Dada la instrucción de Intel 8086:
ADD AL, BL
Indique parejas de valores para AL y BL de forma de lograr las cuatro combinaciones de Carry y Overflow posibles, señalando a cual corresponde cada una.

Pregunta 3

Enumere las entradas de la Unidad de Control y describa la función que la misma debe llevar adelante.

Pregunta 4

Describa los riesgos (obstáculos) que se presentan el uso de una unidad de control con pipeline. Indique un ejemplo de cada riesgo y para solamente uno de ellos indique una posible solución.

Problema 1

Se desea implementar un circuito lógico que indique en su salida la cantidad de ceros o unos consecutivos recibidos en su bit de entrada . La salida se compone de tres bits: el primer bit indica si el último dígito binario en la entrada fue cero o uno, y los últimos dos bits indican, en forma codificada, la cantidad de dicho dígito consecutivo recibido.

Salida codificada	Descripción
00	Más de tres dígitos consecutivos
01	Un dígito consecutivo
10	Dos dígitos consecutivo
11	Tres dígitos consecutivos

Se pide: Implementar el circuito antes descripto utilizando la metodología del curso. Se dispone de flip-flops D y compuertas básicas.

Problema 2

Se plantea a continuación una estructura de datos que implementa un árbol binario de enteros positivos, así como una función que recibe un parámetro con el puntero a un árbol y devuelve dos resultados: la suma de todos los elementos del árbol y una bandera que indica si la suma fue mayor en el sub-árbol derecho o no.

```
typedef struct {
    unsigned short valor;
    nodo* izq, der;
} nodo;

(short, char) sumaYMayor(nodo* arbol){
    if (arbol == NULL)
        return (0, False);
    char descartar;
    unsigned short sumaIzq, sumaDer, sumaTotal;
    (sumaIzq, descartar) = sumaYMayor(arbol->izq);
    (sumaDer, descartar) = sumaYMayor(arbol->der);
    sumaTotal = sumaIzq + sumaDer + arbol->valor;

    if (sumaIzq > sumaDer)
        return (sumaTotal, False);
    else
        return (sumaTotal, True);
}
```

Se pide:

A) Compilar en 8086. Asuma que los punteros son desplazamientos en ES. El parámetro y los resultados se pasan por el stack. No es requerido preservar los registros.

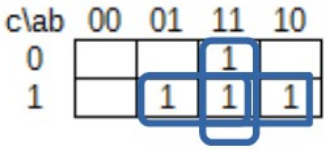
B) Dado un árbol de N nodos, determinar cuál es el consumo de stack en el peor caso.

Respuesta 1

Expresar la función mayoría de tres variables en notación Σ (sigma = suma de productos canónicos) y minimizarla utilizando Karnaugh.

M = Σ(3, 5, 6, 7)

Karnaugh:



M = ab + ac + bc

Respuesta 2

Dada la instrucción de Intel 8086:
ADD AL, BL
Indique parejas de valores para AL y BL de forma de lograr las cuatro combinaciones de Carry y Overflow posibles, señalando a cual corresponde cada una.

AL = 0x00, BL = 0x00 → Carry = 0, Overflow = 0
AL = 0xFF, BL = 0x01 → Carry = 1, Overflow = 0
AL = 0x40, BL = 0x40 → Carry = 0, Overflow = 1
AL = 0x80, BL = 0x80 → Carry = 1, Overflow = 1

Respuesta 3

Enumere las entradas de la Unidad de Control y describa la función que la misma debe llevar adelante.

Las entradas de la Unidad de Control son los bits del Registro de Instrucción (IR) (registro interno de la CPU donde se lee desde memoria la instrucción a ejecutar) y las banderas (o más en general los bits del registro de FLAGS / Processor Status).
La UC, por ser un circuito secuencial, también tiene entradas de reloj y reset.
La función de la UC es llevar a cabo el “ciclo de instrucción”, conjunto de pasos requeridos para ejecutar una instrucción.

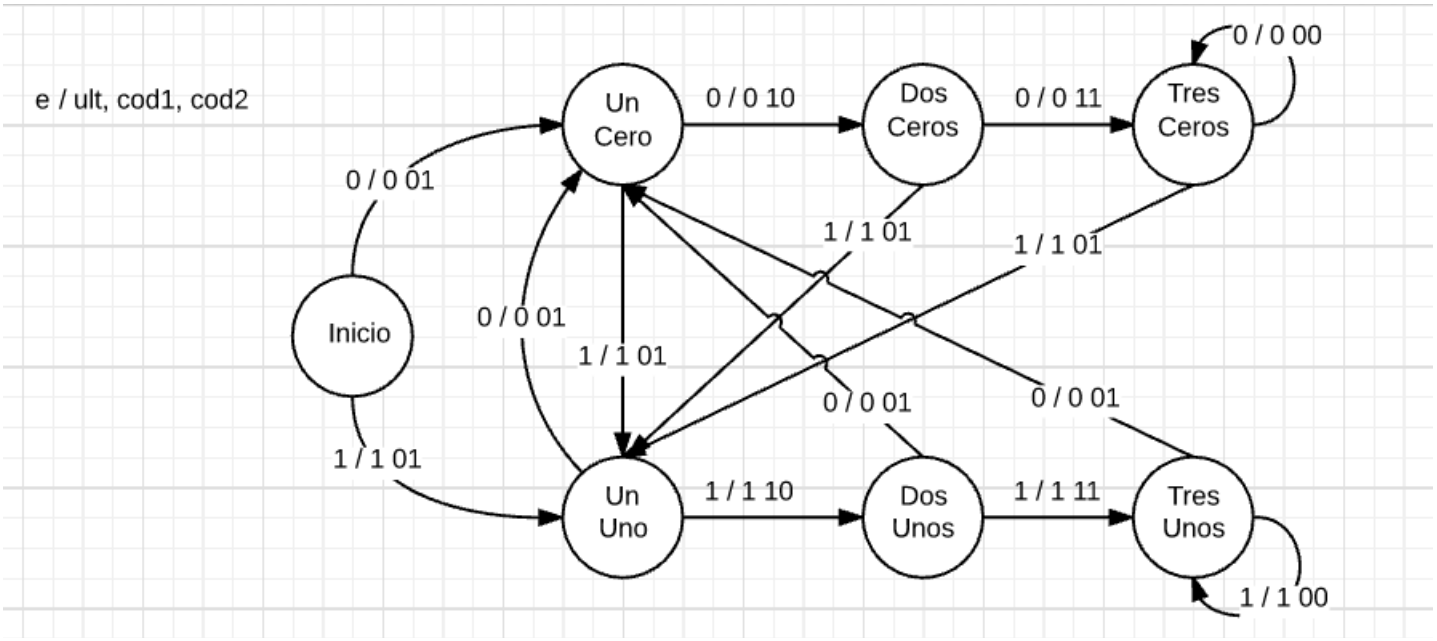
Respuesta 4

Describa los riesgos (obstáculos) que se presentan el uso de una unidad de control con pipeline. Indique un ejemplo de cada riesgo y para solamente uno de ellos indique una posible solución.

Los riesgos (obstáculos, hazards en inglés) se clasifican en: estructurales, de datos y de control.
Un riesgo estructural es el que ocurre cuando dos etapas del pipeline necesitan el mismo recurso de hardware al mismo tiempo. Por ejemplo la etapa de Fetch está intentando leer la próxima instrucción y

la de Read está intentando acceder al operando en memoria de una instrucción ya leída, por lo que ambas deben usar ambas el bus hacia la memoria. Una forma general de solucionar este tipo de riesgos es agregar hardware y una forma de resolver la situación descripta en el ejemplo es untilizar una arquitectura de Harvard (que tiene memorias y buses de memoria separados para datos e instrucciones).
Un riesgo de datos ocurre cuando existe interdependencia de datos entre dos instrucciones que se están ejecutando en el pipeline. Por ejemplo una instrucción posterior necesita como operando el resultado de una instrucción anterior que aún no llegó a actualizar el valor en el lugar de almacenamiento.
Un riesgo de control ocurre como consecuencia de las modificaciones en el puntero de instrucción provocadas por los saltos, condicionales o incondicionales, que generan que la próximas instrucciones a ejecutarse no sean las que están a continuación en memoria (que es lo que asume por defecto la etapa de fetch), con la consiguiente penalización por tener que descartar instrucciones que ya se había comenzado a ejecutar.

Solución Problema 1



Siete estados, se necesitan tres flip flops.

Codificación de Estados (Se eligen de forma tal que el último bit indique si se trata de un estado que cuenta cero o uno, y los primeros dos bits indican cuántos se contaron)

Inicio:	000
Un Cero:	010
Dos Ceros:	100
Tres Ceros.	110
Un Uno:	011
Dos Unos:	101
Tres Unos:	111

Como son flip flops tipo D, la ecuación del próximo estado coincide con la ecuación de la entrada de los flip flops.

Tabla de Verdad:

Estado Actual			Entrada	Próximo Estado (= Entrada de FFs)			Salidas		
q2	q1	q0	e	d2	d1	d0	ult	cod1	cod0
0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	1	1	1	0	1
0	0	1	0	X	X	X	X	X	X
0	0	1	1	X	X	X	X	X	X
0	1	0	0	1	0	0	0	1	0
0	1	0	1	0	1	1	1	0	1
0	1	1	0	0	1	0	0	0	1
0	1	1	1	1	0	1	1	1	0
1	0	0	0	1	1	0	0	1	1
1	0	0	1	0	1	1	1	0	1
1	0	1	0	0	1	0	0	0	1
1	0	1	1	1	1	1	1	1	1
1	1	0	0	1	1	0	0	0	0
1	1	0	1	0	1	1	1	0	1
1	1	1	0	0	1	0	0	0	1
1	1	1	1	1	1	1	1	0	0

Minimizaciones por Karnaugh:

	q2q1\q0e	OO	O1	11	1O
d2	OO	O	O	X	X
	O1	1	0	1	0
	11	1	0	1	0
	1O	1	0	1	0

d2 = q0.e + !q0.!e.q1 +!q0.!e.q2

	q2q1\q0e	OO	O1	11	1O
	OO	1	1	X	X
d1	O1	0	1	0	1
	11	1	1	1	1
	1O	1	1	1	1

d1 = q2 + !q1 + !q0.e + q0.!e

	q2q1\q0e	OO	O1	11	1O
	OO	0	1	X	X
d0	O1	0	1	1	0
	11	0	1	1	0
	1O	0	1	1	0

d0 = e

	q2q1\q0e	OO	O1	11	1O
	OO	O	1	X	X
ult	O1	0	1	1	0
	11	0	1	1	0
	1O	0	1	1	0

ult = e

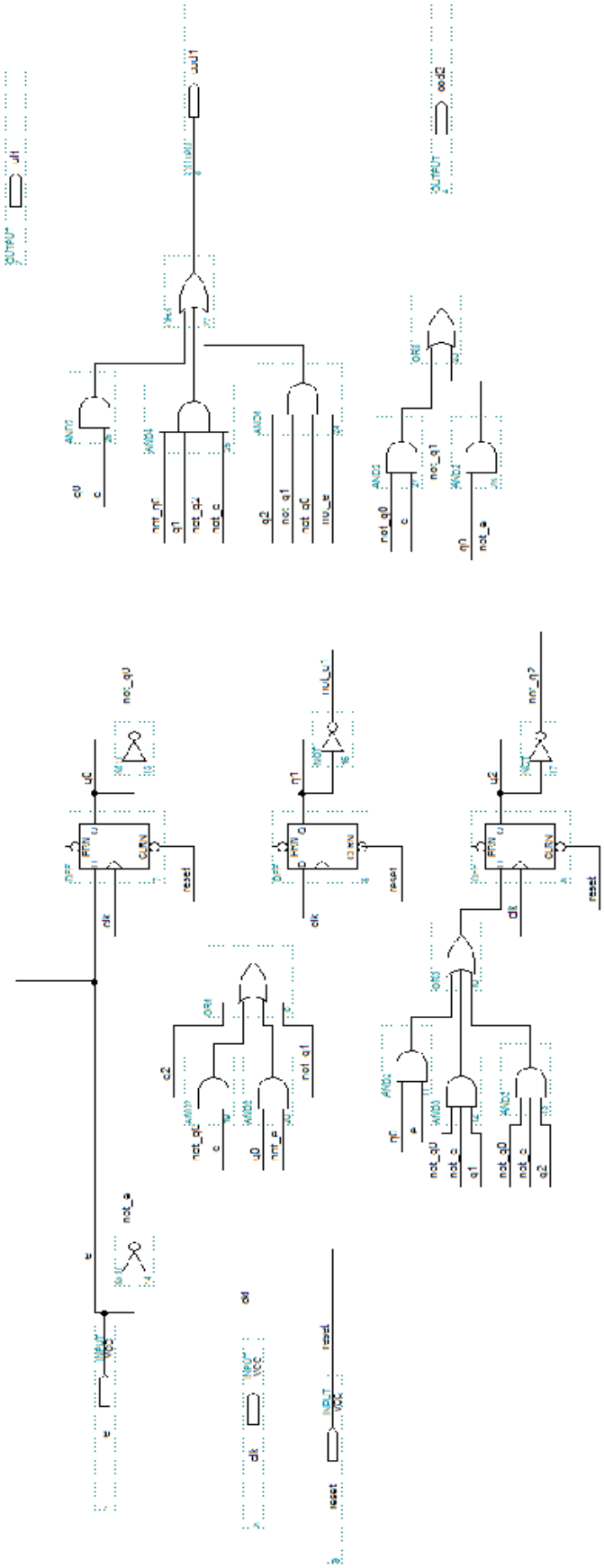
	q2q1\q0e	OO	O1	11	1O
	OO	0	0	X	X
cod1	O1	1	0	1	0
	11	0	0	1	0
	1O	1	0	1	0

cod1 = q0.e + !q2.q1.!e.!q0 + q2.!q1.!e.!q0

	q2q1\q0e	OO	O1	11	1O
	OO	1	1	X	X
cod2	O1	0	1	0	1
	11	0	1	0	1
	1O	1	1	1	1

cod2 = !q0.e + q0.!e + !q1

Circuito:



Solución Problema 2

a)

```
PROC sumaYMayor
   SUB SP,2            ; reservo espacio para segundo parámetro
   PUSH BP
   MOV BP,SP
   PUSH CX
   PUSH AX
   PUSH DX
   PUSH SI
   PUSH DI
   PUSH BX

   XOR AX,AX           ; sumaTotal

   MOV BX,[BP + 6]
   CMP BX,0
   JE FIN

   MOV DX,ES:[BX + 2]
   PUSH DX
   CALL sumaYMayor
   POP SI              ; descartar
   POP CX              ; sumaIzq

   MOV DX,ES:[BX + 4]
   PUSH DX
   CALL sumaYMayor
   POP SI              ; descartar
   POP DI              ; sumaDer

   ADD AX,CX
   ADD AX,DI
   ADD AX,ES:[BX]      ; sumaTotal = sumaIzq + sumaDer + arbol->valor

   CMP CX,DI
   JLE IF_ELSE
   MOV SI,0
   JMP FIN

IF_ELSE:
   MOV SI,1

FIN:
   MOV CX,[BP + 4]
   MOV [BP + 2],CX
   MOV [BP + 4],SI
   MOV [BP + 6],AX

   POP BX
   POP DI
   POP SI
   POP DX
   POP AX
   POP CX
   POP BP
   RET
END PROC
```

b) El consumo de stack en cada llamada es de 20 bytes: 2 de parámetro de entrada, 2 para la dirección de retorno, 2 del espacio reservado para el resultado y 14 de contexto. Para el peor caso, en el que el arbol degenera en una lista, con N elementos en el arbol, se reailzan N llamadas recursivas y un paso base. En total se requieren 20(N + 1) bytes.