Primer parcial de Arquitectura de Computadoras

5 de octubre de 2024

Instrucciones:

- Indique su nombre, apellido y número de cédula en todas las hojas que entregue.
- Las hojas deben estar numeradas y en la primer hoja debe escribirse el total de hojas entregadas.
- Apague su celular. No puede utilizar material ni calculadora.
- La duración del parcial es de tres horas, incluyendo el tiempo para completar sus datos.

Pregunta 1 (3 puntos)

Para las representaciones valor absoluto y signo, complemento a 2 y desplazamiento ($d=2^{n-1}$) de n bits:

a) Indicar el rango de representación.
b) Codificar el cero. Incluir todas las representaciones en caso de tener varias.
c) Codificar el menor y el mayor número representable.

Pregunta 2 (3 puntos)

Dada la función f(a, b, c) = ab + a'bc'

a) Dar la expresión booleana como suma de productos canónicos.
b) Dar la expresión mínima en dos niveles, aplicando Karnaugh.
c) Dar la expresión mínima en dos niveles, aplicando Karnaugh, pero agrupando ceros (producto de sumas).

Pregunta 3 (3 puntos)

Considere una arquitectura de Von Neumann de 16 bits, registros de 16 bits y formato de instrucción:

CÓDIGO de OPERACIÓN (5 bits)	REGISTRO (3 bits)	DIRECCIÓN (8 bits)

Sabiendo que la memoria es direccionable de a byte, indique, fundamentando su respuesta:

a) ¿Cuántos registros como máximo dispone la arquitectura?

b) ¿Cuál es el máximo espacio de memoria direccionable utilizando direccionamiento directo?

c) ¿Cuál es el máximo espacio de memoria direccionable utilizando direccionamiento indirecto por memoria?

Pregunta 4 (3 puntos)

a) Indique los tres componentes principales de una CPU.
b) Para un componente de la parte anterior indique el tipo de circuito, sus entradas y salidas, y el significado de ellas.

Ejercicio 1 (12 puntos)

Se desea extender las funcionalidades del lenguaje C para números racionales. Los números racionales son expresados en punto fijo de 16 bits en total y 6 bits para la parte fraccional. Escribir una función que realice la suma entre dos números racionales y devuelva una palabra de 19 bits con el siguiente formato:

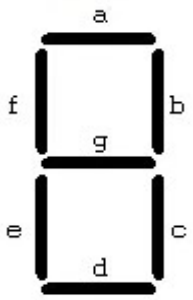
bit 0: bandera Z (cero)
bit 1: bandera V (overflow)
bit 2: bandera N (negativo)
bits 18..3: resultado de la suma (16 bits)

Ejercicio 2 (16 puntos)

Un sistema de números para atender clientes en una panadería consta de un display digital que muestra el número de turno actual, permitiendo a los clientes esperar su turno de manera organizada y eficiente. Se desea utilizar una memoria ROM para convertir “números” que tienen el formato LETRA DIGITO DIGITO. LETRA se codifica en ASCII (8 bits), y puede ser tanto mayúscula como minúscula, mientras que los DIGITOs en BCD empaquetado. La salida de la memoria es conectada a tres displays de siete segmentos que presentan el “número” en un formato amigable.

a) Indique entradas, salidas y organización de la ROM solicitada.
b) A partir de ROMs de 2^{15}x10 y 2^{15}x11 construya la ROM de la parte a).
c) Implemente un programa en C que cargue el contenido de la ROM. En caso de recibir una entrada inválida, la salida de la ROM debe ser tal que el display muestre “Err”.

Nota: se dispone de las funciones *char charTo7Seg(char ascii)* y *char bcdTo7Seg(char bcd),* que dados un carácter o un bcd devuelve su codificación en siete segmentos (el display consta de los segmentos de **a** ... **g**, el segmento **a** corresponde al bit 0 y el **g** al bit 6).



Respuesta Pregunta 1

a) Los rangos de las representaciones son:

- Valor absoluto y signo: $-(2^{n-1} - 1) \leq X \leq 2^{n-1} - 1$
- Complemento a 2: $-2^{n-1} \leq X \leq 2^{n-1} - 1$
- Desplazamiento: $-2^{n-1} \leq X \leq 2^{n} - 1 - 2^{n-1}$

b) Los representaciones del cero son:

- Valor absoluto y signo: tiene dos representaciones 1) $1\overbrace{0...0}^{n-1}$ y 2) $0\overbrace{0...0}^{n-1}$
- Complemento a 2: tiene una sola representación: $\overbrace{00...0}^{n}$
- Desplazamiento: tiene una sola representación: $1\overbrace{0...0}^{n-1}$

c) Los codificaciones del menor número representable son:

- Valor absoluto y signo: $\overbrace{11...1}^{n}$
- Complemento a 2: $1\overbrace{0...0}^{n-1}$
- Desplazamiento: $\overbrace{00...0}^{n}$

Los codificaciones del mayor número representable son:

- Valor absoluto y signo: $0\overbrace{1...1}^{n-1}$
- Complemento a 2: $0\overbrace{1...1}^{n-1}$
- Desplazamiento: $\overbrace{11...1}^{n}$

Respuesta Pregunta 2

a)

Primero hacemos la tabla de verdad de la función f:

a	b	c	a.b + a'.b.c'	
0	0	0	0	
0	0	1	0	
0	1	0	1	⇨ $\bar{a}.b.\bar{c}$
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	1	⇨ $a.b.\bar{c}$
1	1	1	1	⇨ $\bar{a}.\bar{b}.\bar{c}$

Por lo tanto, la expresión booleana como suma de productos canónicos de la función f está dada por:
$f(a,b,c) = \bar{a}.b.\bar{c} + a.b.\bar{c} + a.b.c$

b)

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

$f = a.b + b.\bar{c}$

c)

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

$f = b.(a+\bar{c})$

Respuesta Pregunta 3

a) Dado que el formato de instrucción indica que se usan tres bits para los registros, el máximo número de registros (visibles) que dispone la arquitectura es $2^3 = 8$ registros.

b) Cuando se utiliza direccionamiento directo, la dirección de memoria se incluye en la propia instrucción. Dado que el campo para especificar la dirección de memoria tiene 8 bits, se pueden direccionar hasta $2^8 = 256$ posiciones. El máximo espacio de direccionamiento entonces es 256 posiciones en el rango [0,255].

c) Cuando se utiliza direccionamiento indirecto por memoria, la dirección incluida en la instrucción contiene la dirección de memoria donde se almacena la dirección del dato. La memoria consta de palabras de 16 bits (por tratarse de una arquitectura de 16 bits), por lo cual se pueden almacenar direcciones de hasta $2^{16} = 65536$ posiciones. El máximo espacio de direccionamiento entonces es 65536 posiciones en el rango [0,65535].

Respuesta Pregunta 4

a) Los tres componentes principales de una CPU son: la Unidad Aritmético Lógica (ALU), la Unidad de Control (UC) y el Banco de Registros.

b) Nota: aquí describimos los tres componentes, solo se requería describir uno.

ALU
La ALU es un circuito combinatorio
Entradas:

- opcode: indica la operación a realizar
- op1: uno de los operandos de la operación a realizar
- op2: el otro operando

Salidas:
- resultado: el resultado de la operación realizada
- flags: son las banderas modificadas acorde a la operación realizada

Banco de Registros
El banco de registros es un circuito secuencial
Entradas:
- Identificador de los registros a leer o escribir
- Valor a ser escrito en el registro seleccionado

Salidas:
- Valor leído de los registros seleccionados.

Unidad de Control
La unidad de control es un circuito secuencial
Entradas:
- flags: son las banderas generadas por la ALU
- IR: es el registro de instrucción que contiene la instrucción a ser ejecutada (leída de memoria) y le permite a la UC conocer la operación a realizar, los operandos a utilizar y su ubicación.
- señales de control (p.e. INT)

Salidas:
- en general señales que permiten controlar la interconexión de los distintos componentes
- señales de control del acceso al banco de registros y a otros registros auxiliares
- señales de control de la operación a realizar en la ALU
- señales de control de los accesos al bus de control del sistema (RD, WR, entre otros)

Solución Problema 1

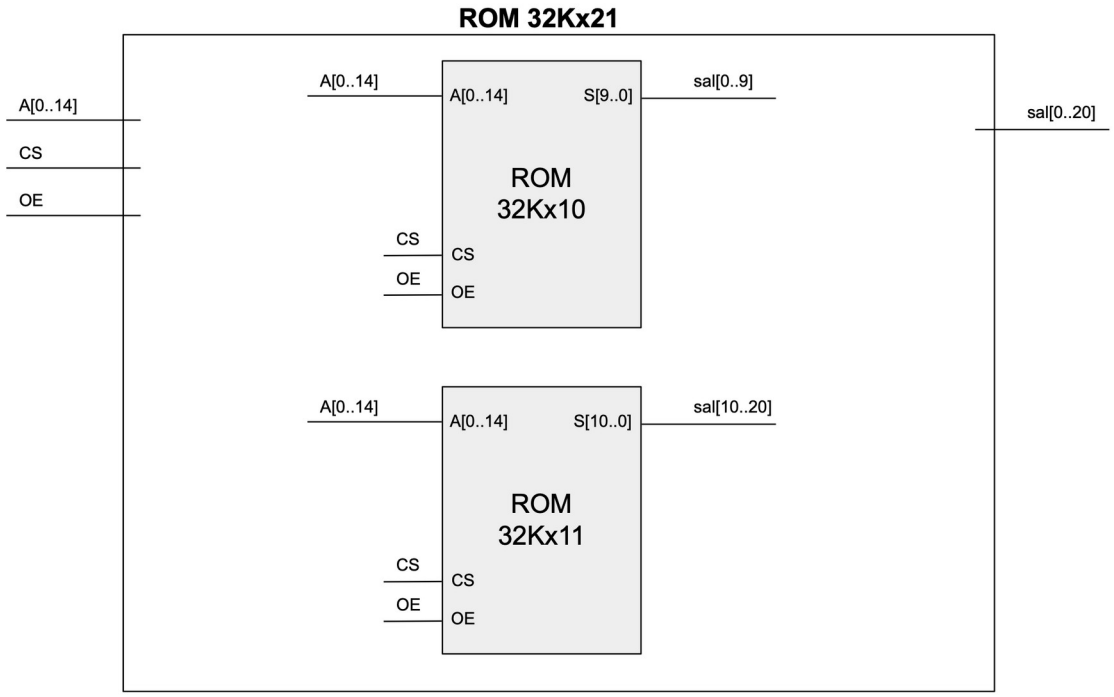
```
int suma (short num1, short num2) {
        char flag_Z = 0;
        char flag_V = 0;
        char flag_N = 0;
        int resultado = num1 + num2;
        if (num1 * num2 > 0) {          // los números tienen el mismo signo
           if (num1 * resultado < 0) {  // y el resultado signo opuesto
              // hay overflow
              flag_V = 1;
           }
        }
        if (resultado & 0xFFFF == 0)
           flag_Z = 1;
        if (resultado & 0x8000 != 0)
           flag_N = 1;
        return (0xFFFF & resultado) << 3 | flag_N << 2 | flag_V << 1 |
               flag_Z);
}
```

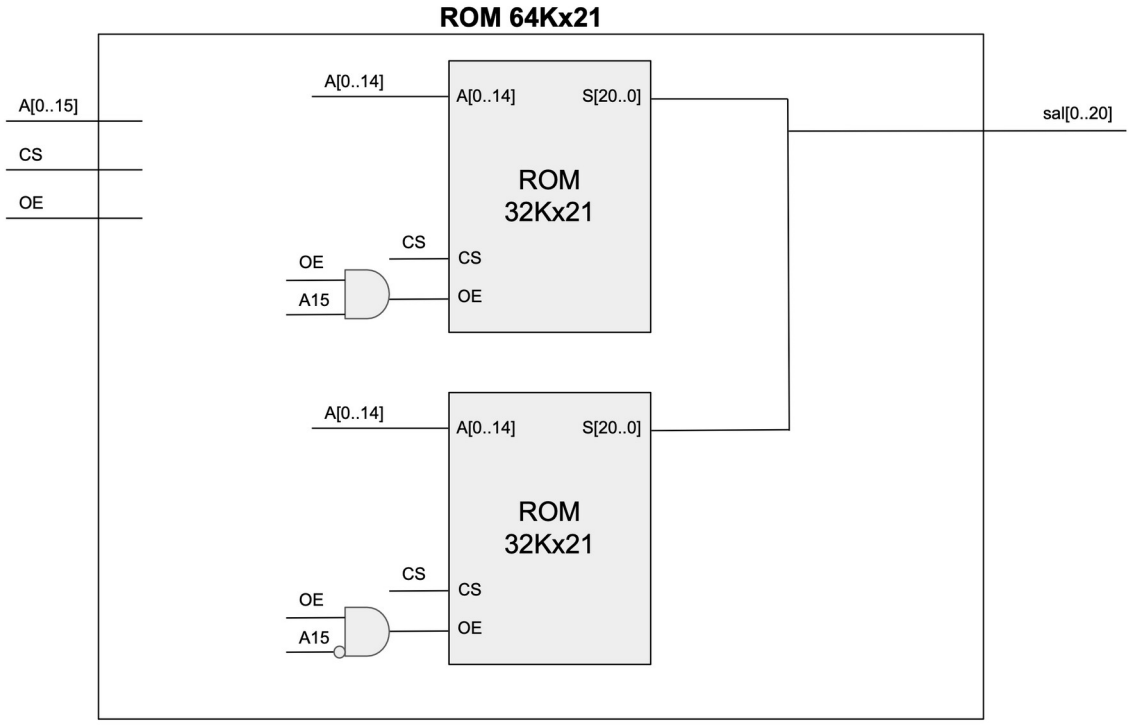
Observación: recordar que en la representación punto fijo se opera directamente con las representaciones y la suma es consistente

Solución

a) Como entradas tenemos una letra en ASCII y dos dígitos en BDC empaquetado. La letra son 8 bits mientras que cada dígito son 4 bits, por lo que tenemos 8+4+4 entradas, es decir 16 entradas.
La salida serán los tres displays de 7 bits cada uno, sumando un total de 7x3 salidas, es decir 21 salidas.
La organización de la ROM es 2^{16}x21, es decir 64Kx21

b) Primero construimos una ROM de 32Kx21 usando una ROM de 32Kx10 y una ROM de 32Kx11. Luego, usamos dos ROMs de 32Kx21 para construir la ROM de 64Kx21.





c)

```
int ROM[65536];

void cargaROM () {
         for (int i = 0; i < 65536; i++) {
            char letra = i >> 8;
            char digito1 = (i >> 4) & 0xF;
            char digito0 = i & 0xF;
            if ( !((letra >= 'A' && letra <= 'Z') || (letra >= 'a' && letra <= 'z'))
                 || digito1 > 9 || digito0 > 9) {
               ROM[i] = charTo7Seg('E') <<14 | charTo7Seg('r')<<7 |
                           charTo7Seg('r');
            } else {
               ROM[i] = charTo7Seg(letra)<<14 | bcdTo7Seg(digito1)<<7 |
                                    bcdTo7Seg(digito0);
            }
         }
}
```