Universidad Autónoma de Baja California

Ingeniería en Computación

Facultad de Ciencias Químicas e Ingeniería

Organización y Arquitectura de Computadoras

Practica 2

Zavala Román Irvin Eduardo

Grupo: 551

01/09/2021

Periodo 2021-2

Objetivo

Identificar la organización de una computadora de propósito general, para comprender su estructura y funcionamiento, analizando la interconexión de sus componentes básicos, con una actitud crítica, propositiva y visionaria.

Desarrollo

Responda los siguientes cuestionamientos acerca del simulador MARIE.

1. Complete la Tabla 1 describiendo con sus propias palabras las acciones que realizan cada una de las instrucciones del simulador.

2. Complete la Tabla 2 indicando la función de cada uno de los registros.

3. ¿Qué tamaño en bits tiene el bus de direcciones?

4. ¿Qué tamaño en bits tiene el bus de datos?

5. ¿Qué tamaño en bits tiene el código de operación (opcode) de las instrucciones?

6. ¿Cuál es la dirección máxima de memoria que se puede acceder?

7. ¿Por qué el código de operación de los registros es de 3 bits?

8. ¿Por qué el registro MAR es de 12 bits?

9. ¿Por qué el registro MBR es de 16 bits?

10. Escriba un programa que contenga la subrutina Division, la cual recibe dos números en las variables A y B y almacena en la variable R el resultado de A/B. En el código principal, solicite al usuario dos números y despliegue la división del primero entre el segundo.

11. Escriba un programa que contenga la subrutina Pares, la cual despliega en pantalla los primeros N números pares. N es un número ingresado por el usuario en el código principal.

12. El plan de una compañía celular incluye 100 minutos de llamadas nacionales y 70 mensajes de texto por $80 al mes. Cada minuto adicional de llamadas nacionales cuesta $1.00, cada mensaje adicional $2.00 y las llamadas internacionales se cobran a $3 el minuto. Escriba un programa que solicite al usuario tres números que representan los minutos usados en llamadas nacionales e internacionales, y los mensajes enviados. Finalmente, calcule y despliegue el total a pagar.

Resultados

1.

Introducción	Descripción
Add X	Añade el valor del acumulador en la dirección X
Subt X	Resta el valor del acumulador en la dirección X
Addl X	Suma el valor de la dirección del valor X al acumulador
Clear	El acumulador se hace 0
Load X	Carga contenido de la dirección X en el acumulador
Store X	Guarda el contenido del acumulador en la dirección X
Input	Pide entrada de datos
Output	Imprime el valor del acumulador
Jump X	“Salta” a la dirección X
Skipcond(c)	Salta la instrucción dependiendo del valor de C
JnS X	Guarda el contador de programa en la dirección X y pasa a la dirección siguiente, se usa para subrutinas
Jumpl X	“Salta” al valor de X, no la dirección
Storel	Guarda el valor del acumulador en la dirección indirecta.
Loadl	Carga el valor de la dirección indirecta en el acumulador
Halt	Termina el programa

2.

Registro	Funcion
MAR	Guarda los valores en la dirección dada
PC	Apunta a la siguiente dirección que ejecutara el CPU
MBR	Guarda los valores cuando se transfieren
AC	Acumulador, retiene resultados
IN	Puertos de entrada
OUT	Puertos de salida
IR	Registro de instruccion, retiene la instruccion actual

3. 12 bits
4. 16 bits
5. 4 bits
6. 4096 direcciones
7. Porque hay un set de 8 registros de control, que se pueden generar con 3 bits
8. Porque ahí se guarda la dirección de los datos que se referencian, las direcciones son de 12 bits como se ve en las instrucciones
9. Como la máquina es de 16 bits, el MBR tiene ese valor
10. INPUT

    Store A

    INPUT

    Store B

    Load r

    Add A

    Store r

    if, Load B

        Subt A

```
        Skipcond 800
    Jump Division
    Load A
    Output//Lo que salga de aqui en realidad es 0.a
Halt
Division, Load q
                Add one
        Store q
        Load r
        Subt B
        Store r
        Load B
        Subt r
        Skipcond 800
        Jump Division




Load q  /Este el lado antes del punto
Output
Load r  /Despues del punto
/q.r
Output
Halt
```

```
aux,DEC 0
aux2, DEC 0
r, DEC 0
A, DEC 0
B, DEC 0
one, DEC 1
num, DEC 0
q, DEC 0
```

11. INPUT

```
Store A
Pares,    load num
          Add two
          Store num
          Output
          Load A
          Subt one
          Store A
          Skipcond 400
          Jump Pares
Halt
A, DEC 0
two, DEC 2
one, DEC 1
```

num, DEC 0

12. INPUT

    Store nacionales

    INPUT

    Store internacionales

    INPUT

    Store enviados

    Load nacionales

    Subt cien

    Store nacionales

    Load enviados

    Subt setenta

    Store enviados

    load total_final

    Add ochenta

    Store total_final

    load nacionales

    Store nacionales

    Skipcond 800

    Jump enviados_extras

    nacionales_extras, load total_nacionales

    Add uno

    Store total_nacionales

```
                load nacionales
                Subt uno
                Store nacionales
                Skipcond 400
                Jump nacionales_extras


enviados_extras,   load enviados
                Skipcond 800
                Jump total_internacional
                                Load total_enviados
                                Add dos
                Store total_enviados

                load enviados
                Subt uno
                Store enviados
                Skipcond 400
                Jump enviados_extras


total_internacional, Load total_inter
                                        Add tres
                 Store total_inter

                 Load internacionales
```

```
        Subt uno
        Store internacionales
        Skipcond 400
        Jump total_internacional


load total_final
Add total_inter
Store total_final
load total_final
Add total_nacionales

Store total_final
load total_final
Add total_enviados

Store total_final
Output

Halt


uno, DEC 1
tres, DEC 3
dos, DEC 2
setenta, DEC 70
ochenta, DEC 80
```

cien, DEC 100

nacionales, DEC 0

internacionales, DEC 0

enviados, DEC 0

nacional_incluido, DEC 100

mensajes_incluidos, DEC 70

total_inter, DEC 0

total_nacionales, DEC 0

total_enviados, DEC 0

total_final, DEC 0

/Al ejecutar salen “breakpoints”, no se que es pero si le da continue acaba la /ejecución xd

Conclusiones y comentarios

El ver como funciona un CPU a un nivel tan bajo es impresionante ya que este tipo de procesos suceden en las computadoras en cada momento y nos dan el acceso a un mundo tan grande como es la computación.

Dificultades en el desarrollo

Los últimos 3 programas se me hicieron complicados no por la dificultad de los problemas sino que trataba de traducir de lenguajes de programación que ya conocía a este y no siempre salía bien.