

Organización de la computadora y diseño de arquitectura para el rendimiento

ORGANIZACIÓN Y ARQUITECTURA DE COMPUTADORES

Freddy L. Abad L., Bryan A. Aguilar Y., Christian X. Collaguazo M.

Escuela de Informática, Facultad de Ingeniería, Universidad de Cuenca, Ecuador

[freddy.abadl,bryan.aguilar,christian.collaguazo}@ucuenca.edu.ec](mailto:{freddy.abadl,bryan.aguilar,christian.collaguazo}@ucuenca.edu.ec)



1.1 Debe escribir un programa IAS para calcular los resultados de la siguiente ecuación.

$$Y = \sum_{X=1}^N X$$

Suponga que el cálculo no produce un desbordamiento aritmético y que X, Y y N son enteros positivos con $N \geq 1$.

$$\text{Sum}(Y) = \frac{N(N+1)}{2}$$

A. Use la ecuación para escribir el programa en IAS.

La computación no produce desbordamiento aritmético. Las variables X, Y y N son números enteros positivos donde el valor de N es mayor o igual a 1.

Programa IAS

Ubicación	Instrucción/Valor	Comentarios
0	\diamond	Constant (N) [inicializado a algún valor]
1	1	Constant; Integer value=1
2	2	Constant; Integer value=2
3	0	Variable Y (Inicializada a cero entero); Sum(Y)
4L	LOAD M(0)	N -> AC
4R	ADD M(1)	AC+1 -> AC
5L	MUL M(0)	N(N+1) -> AC
5R	DIV M(2)	AC/2 -> AC
6L	STORE M(3)	AC -> Y, guardando la suma en la variable Y

6R	JUMP M(6,20:39)	Done, HALT (Detener)
----	-----------------	----------------------

- Una instrucción IAS es una instrucción de 40 bits y se divide en la parte izquierda con 20 bits y la parte derecha con 20 bits.
- El código inicializa la constante N con algún valor y se encuentra en la ubicación 0, posteriormente se inicializa con un valor entero de 2.
- La variable Y se inicializa al valor entero 0.
- La constante N se almacena en el acumulador AC. ($N \rightarrow AC$) y esto se almacena en la parte izquierda de la 4ta ubicación.

Toma un acumulador incrementando el valor en 1 y guardarlo en otro acumulador, es decir $N+1 \rightarrow AC$ este se almacena en la parte derecha de la 4ta ubicación.

Este proceso definimos a continuación:

Instrucción Izquierda	Instrucción Derecha		
LOAD M(0)	$N \rightarrow AC$	ADD M(1)	$N+1 \rightarrow AC$

Multiplicar el acumulador con la constante N y guardarlo en el Acumulador, es decir, $N(N+1) \rightarrow AC$, se almacena en la parte izquierda de la 5ta instrucción.

Dividir el acumulador por 2 y guardarlo en el acumulador, esto se almacena en la parte derecha de la 5ta ubicación.

Este proceso se detalla a continuación:

Instrucción Izquierda	Instrucción Derecha		
MUL M(0)	$N(N+1) \rightarrow AC$	DIV M(2)	$(AC/2) \rightarrow AC$

Finalmente, se guarda en el acumulador en la variable Y

Este proceso definimos a continuación:

Instrucción Izquierda	Instrucción Derecha		
STORE M(3)	$AC \rightarrow Y$	JUMP M(6, 20:39)	HALT

B. Hágalo de la “manera difícil”, sin usar la ecuación de la parte (a).

Ubicación	Instrucción/Valor	Comentarios
0	\diamond	Constant (N) [inicializado a algún valor]
1	1	Constant (Incremento del contador de bucles)

2	1	Variable i (valor de índice de bucle; actual)
3	1	Variable Y = Sum(X) (Inicializados en uno)
4L	LOAD M(0)	N -> AC (el límite máximo)
4R	SUB M(2)	Calcular N-i -> AC
5L	JUMP + M(6, 0:19)	Comprobar AC>0 [i< N]
5R	JUMP + M(5, 20:39)	i=N, done so HALT
6L	LOAD M(2)	i
6R	ADD M(1)	i+1 en AC
7L	STOR M(2)	AC -> i
7R	ADD M(3)	i+Y in AC
8L	STOR M(3)	AC -> Y
8R	JUMP M(4, 0:19)	Continuar en la instrucción ubicada en la dirección 4L

Explicación:

- Tome una constante N que indica el número máximo
- Hacer la constante como un contador de bucle
- Tome una variable i que es el índice de bucle
- La variable Y se establece como la suma de todos los valores X establecidos inicialmente como 1
- Cargue el valor N en el acumulador y mantenerlo en la parte izquierda de la 4ta instrucción.
- Restar el valor i de la N y guardarlo en el acumulador

Este proceso definimos a continuación:

Instrucción Izquierda	Instrucción Derecha		
LOAD M(0)	N -> AC	SUB M(2)	N - i -> AC

Si el valor de CA es mayor a 0, además si i es inferior a N a la parte izquierda de la 6ta instrucción. Si el valor de N es igual a i, HALT y saltar a la parte derecha de la 5ta instrucción.

En una instrucción 0:19, indique la porción izquierda 20 bits y 20:39 indica la porción derecha 20 bits.

Este proceso definimos a continuación:

Instrucción Izquierda	Instrucción Derecha		
JUMP + M(6,0:19)	AC -> 0? [i<N]	JUMP + M(5,20:39)	i=N

Si el valor i es menor que la carga N en la memoria, guárdelo en la parte izquierda de la 6ta instrucción, incrementar el valor i en 1 y guardarlo en el acumulador, guardarlo en la parte derecha de la 6ta instrucción.

Este proceso definimos a continuación:

Instrucción Izquierda	Instrucción Derecha		
LOAD M(2)	i< N	ADD M(1)	i+1 -> AC

- Asignar el valor del acumulador a i y guardarlo en la parte izquierda de la 7ma instrucción.
- Agregar las variables i y Y y guardarlas en el acumulador

Este proceso definimos a continuación:

Instrucción Izquierda	Instrucción Derecha		
STOR M(2)	AC -> i	ADD M(3)	i+Y-> AC

Asignar el valor del acumulador a la variable Y, saltar a la parte izquierda de la 4ta instrucción de igual manera y continuar al final del bucle.

Este proceso definimos a continuación:

Instrucción Izquierda	Instrucción Derecha		
STOR M(3)	AC -> Y	JUMP M(4, 0:19)	LOOP a 4L

1.2 A. En el IAS, ¿cómo sería la instrucción del código de máquina para almacenar los contenidos de un acumulador en la dirección de memoria 8?

Opcode: 00001000

Operando: 000000000010

B. En el IAS, ¿cómo se vería la instrucción de código de máquina para agregar el contenido de la dirección de memoria 16 al acumulador?

Se necesitan dos instrucciones:

1. Cargar el valor de la dirección de memoria 16.
2. La segunda operación es realizar la adición en el acumulador.

Opcode: 00010000

Operando: 000000000010

1.3 El IAS funciona mediante la ejecución repetitiva de un ciclo de instrucción, que consta de dos subciclos: un ciclo de búsqueda y un ciclo de ejecución. En el IAS, describa las tareas realizadas durante el ciclo de captación y las realizadas durante el ciclo de ejecución.

El IAS opera ejecutando repetidamente un ciclo instrucción, como se puede ver en la Figura. Cada ciclo instrucción consta de dos subciclos. Durante el **ciclo de captación**, el codop (código de operación) de la siguiente instrucción es cargado en el IR (registro de instrucción) y la parte que contiene la dirección es almacenada en el MAR (registro de dirección de memoria). Esta instrucción puede ser captada desde el IBR (registro temporal de instrucción), o puede ser obtenida de la memoria cargando una palabra en el MBR (Registro intermedio de memoria), y luego en IBR, IR y MAR. Una vez que el codop está en el IR, se lleva a cabo el **ciclo de ejecución**. Los circuitos de control interpretan el codop y ejecutan la instrucción enviando las señales de control adecuadas para provocar que los datos se transfieran o que la ALU realice una operación.

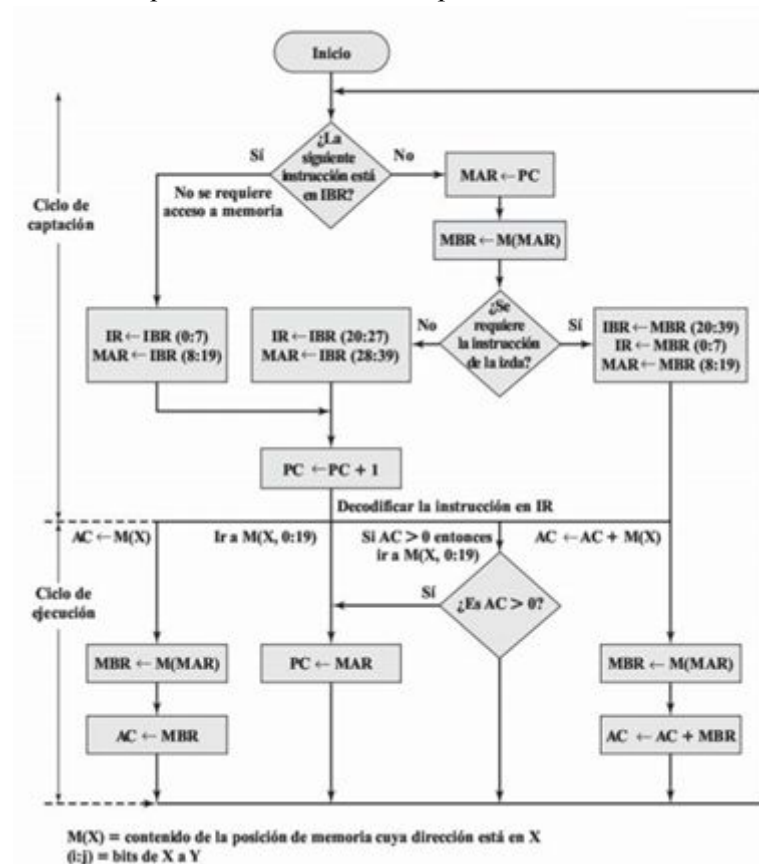


Fig. Diagrama de flujo parcial de operaciones IAS

1.4 Dado el contenido de memoria de la computadora IAS que se muestra abajo

Dirección	Contenido
08A	010FA210FB
08B	010FA0F08D
08C	020FA210FB

Muestre el código del lenguaje ensamblador para el programa, comenzando en la dirección 08A. Explique lo que hace este programa.

Dirección	Contenido
08A	LOAD M(0FA) STOR M(0FB)
08B	LOAD M(0FA) JUMP +M(08D)
08C	LOAD -M(0FA) STOR M(0FB)

Este programa sirve para almacenar el valor absoluto del contenido en la ubicación de memoria 0FA en la ubicación de memoria 0FB.

1.5 Durante el ciclo de búsqueda en la Figura 1.8, ¿por qué siempre se toma una instrucción del IBR?

Esto se da porque el IBR almacena la instrucción que está en IR y parte de la dirección que está cargada en MAR, por lo que si el IBR contiene la siguiente instrucción, en caso de tenerla se evita el acceso a memoria para obtener los valores de IR Y MAR.

1.6 En los modelos IBM 360 65 y 75, las direcciones se dividen en dos unidades de memoria principal separadas (por ejemplo, todas las palabras con números pares en una unidad y todas las palabras con números impares en otra). ¿Cuál podría ser el propósito de esta técnica?

Al tener separado registros pares e impares se pueden acceder hasta dos palabras, gracias a esto aumenta la velocidad y eficiencia del sistema, logrando tener un acceso paralelo a los registros por un solo reloj lo que da una mejor velocidad de procesamiento.

1.7 El rendimiento relativo del IBM 360 Model 75 es 50 veces mayor que el del 360 Modelo 30, pero el tiempo del ciclo de instrucción es solo 5 veces más rápido. ¿Cómo explicas esta discrepancia?

La discrepancia se puede explicar al notar que otros componentes del sistema, aparte de la velocidad del reloj, hacen una gran diferencia en la velocidad general del sistema. En particular, los sistemas de memoria y los avances en el procesamiento de E / S contribuyen a la relación de rendimiento. Un sistema es tan rápido como su enlace más lento. En los últimos años, los cuellos de botella han sido el rendimiento de los módulos de memoria y la velocidad del bus. Al notar la discrepancia de los componentes del sistema se puede explicar.

1.8 Mientras navegas en la tienda de computadoras de Billy Bob, escuchas a un cliente preguntándole a Billy Bob cuál es la computadora más rápida en la tienda que puede comprar. Billy Bob responde: "Estás mirando a nuestros Macintosh. La Mac más rápida que tenemos funciona a una velocidad de reloj de 1.2 GHz. Si realmente quieres la máquina más rápida, deberías comprar nuestro Intel Pentium IV de 2.4 GHz en su lugar ". ¿Es correcto Billy Bob? ¿Qué dirías para ayudar a este cliente?

No, Billy Bob no es correcto, el rendimiento de la computadora no es medido usando solo la velocidad del reloj, y otros factores de los componentes del sistema deben ser considerados, como los

componentes del sistema (memoria, buses, arquitectura) y los conjuntos de instrucciones. Una medida más precisa es ejecutar ambos sistemas en un punto de referencia. Existen programas de referencia para ciertas tareas, como ejecutar aplicaciones de oficina, realizar operaciones de punto flotante, etc. El sistema se puede comparar entre sí por el tiempo que demora en completar estas tareas.

1.9 El ENIAC, un precursor de la máquina ISA, era una máquina decimal, en la que cada registro estaba representado por un anillo de 10 tubos de vacío. En cualquier momento, solo un tubo de vacío estaba en el estado ENCENDIDO, que representa uno de los 10 dígitos decimales. Suponiendo que ENIAC tuviera la capacidad de tener múltiples tubos de vacío en el estado ON y OFF simultáneamente, ¿por qué esta representación es "inútil" y qué rango de valores enteros podríamos representar usando los 10 tubos de vacío?

La representación es inútil ya que representamos solo un solo dígito decimal, en caso de poder usar los 10 tubos de vacío podemos representar números de 0 a 1023.

1.10 Para cada uno de los siguientes ejemplos, determine si se trata de un sistema integrado, explicando por qué o por qué no.

A. ¿Están integrados los programas que entienden la física y / o el hardware? Por ejemplo, ¿uno que usa métodos de elementos finitos para predecir el flujo de fluido sobre las alas de los aviones?

No, ya que no forman parte integral del sistema.

B. ¿El microprocesador interno que controla una unidad de disco es un ejemplo de un sistema integrado?

Sí, ya que independientemente de para qué se utiliza la unidad de disco. El software dentro de la unidad de disco controla el hardware..

C. Los controladores de E / S controlan el hardware, ¿la presencia de un controlador de E / S implica que la computadora que ejecuta el controlador está integrada?

No, ya que la computadora puede usarse para algún propósito general, lo implica que no es parte de un sistema más grande.

D. ¿Es un PDA (Asistente digital personal) un sistema integrado?

No, ya que el software y SO se mantienen en una memoria volátil, la cual no está controlando un hardware en especial.

E. ¿El microprocesador que controla un teléfono celular es un sistema integrado?

Sí, ya que su firmware está controlando el hardware del dispositivo móvil.

F. ¿Se consideran embebidos las computadoras en un gran radar de red de fases? Estos radares son edificios de 10 pisos con uno a tres parches radiantes de 100 pies de diámetro en los lados inclinados del edificio.

Sí, ya que se consideran parte de un solo sistema el cual es controlado por un microprocesador el cual se encarga de controlar el hardware, en este caso de los radares.

G. ¿Se considera integrado un sistema de gestión de vuelo (FMS) tradicional en la cabina de un avión?

Si, ya que en la actualidad el sistema de gestión de vuelo se encarga de realizar tareas de forma independiente, así como permite el manejo de forma manual del sistema. Esto es controlado por la llamada computadora de vuelo.

H. ¿Son las computadoras hardware-in-the-loop (HIL) un simulador embebido?

Si, ya que se encargan de la simulación electrónica de los diferentes sensores. El sistema que se encuentra integrado permite la creación de algoritmos de control.

I. ¿La computadora que controla un marcapasos en el pecho de una persona es un computador embebido?

Si, este se encuentra dentro del cuerpo de una persona y se encarga de generar impulsos eléctricos los cuales no se generan de forma programada, si no en base a monitoreos que realiza el PSA sistema de análisis de marcapasos.

J. Es la computadora que controla la inyección de combustible en un motor de automóvil una computadora embebida?

Si, ya que es una parte importante del sistema automotriz, encargándose del cálculo de la cantidad de combustible a inyectar dentro del motor, todo esto siguiendo una serie de parámetros.