

1. Dado el siguiente patrón de bits:

H'8020 0000

a) ¿Cuál sería el valor decimal del número si correspondiese a un número entero representado en signo y magnitud?

H'8020 0000 → 1000 0000 0010 0000 0000 0000 0000 0000

Como el número está representado en signo y magnitud; el primer bit es el signo y el resto el valor absoluto del número entero, con lo que:

$$N = - 2^{21} = - 2_{1097.152}$$

Resultado	- 2 _{1097.152}
-----------	-------------------------

b) ¿Cuál sería el valor decimal del número si correspondiese a un número real representado en IEEE754?

H'8020 0000 → 1000 0000 0010 0000 0000 0000 0000 0000

Signo: s=1 (N<0)

Bits del exponente sesgado: e= 0000 0000

Bits de la mantisa: 010 0000 0000 0000 0000 0000

Como el exponente sesgado es cero, el número es de-normalizado con lo que, E=-126 y la mantisa esta de-normalizada; es decir, corresponde a 0,01; con lo que:

$$N = - 0,01 \cdot 2^{-126} = -2,9387 \cdot 10^{-39}$$

Resultado	-2,9387·10 ⁻³⁹
-----------	---------------------------

c) ¿A qué dos instrucciones de CODE2 correspondería caso de que fuese parte de un programa de dicho procesador?

Como las instrucciones de CODE-2 son de 16 bits, el patrón corresponderá a 2 instrucciones:

8020 → NAND r0,r2,r0

0000 → LD r0, [00]

Resultado	NAND r0,r2,r0 LD r0, [00]
-----------	------------------------------

2. Hacer un programa en código máquina para CODE-2 que encuentre el número más pequeño dentro de una zona determinada de su memoria y lo proporcione por el puerto de salida OP02. Las direcciones iniciales y finales de la zona las debe dar el usuario a través del puerto de entrada IP01.

Para la realización del programa, considere los siguientes pasos:

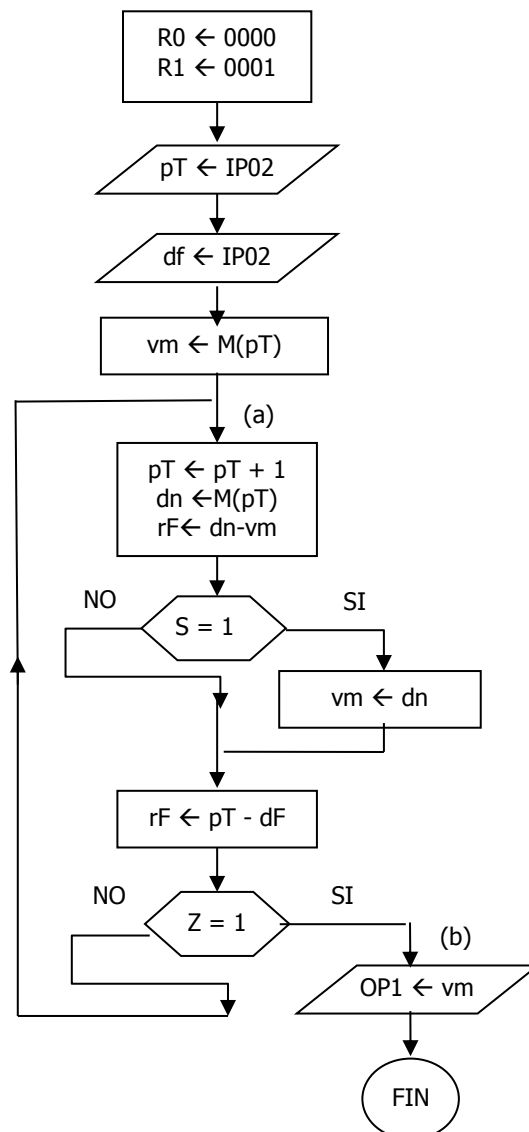
- Organigrama
- Asignación de registros y memoria
- Redactar el programa en nemónicos (código máquina).
- Escribir el programa completo en hexadecimal
- Obtener el tiempo que tardaría en ejecutarse el programa en función del número de posiciones de la zona de memoria a analizar, suponiendo que la frecuencia de reloj de CODE-2 fuese de 10 GHz

a. Organigrama.

Utilizamos las siguientes variables:

- Puntero de memoria dentro de la zona donde se encuentran los números: pT . Su valor inicial será la dirección de comienzo de la zona de datos (di).
- Dirección final de la zona de números: df
- Valor mínimo: vm . Su valor inicial será el del primer número de la zona.
- Dato nuevo a considerar: dn . Este dato se irá leyendo iterativamente de la memoria.

En la figura se encuentra el organigrama.



b. *Asignación de registros y de memoria.*

- El programa se carga a partir de la dirección M(0000)

Rtro.	Contenido	Valor inicial
r0	Valor 0	0000
r1	Valor 1	0001
r3	Puntero de la lista de datos (pT)	En IP01
r4	Dirección final de la lista de datos (dF)	En IP01
r5	Dato nuevo en consideración (dn)	
r6	Valor mínimo (vm)	

a) *Programa en nemónicos y código máquina.* Puede verse en la siguiente tabla:

Dcc. de salto	Dirección	Nemónico	Hexadecimal	Comentarios
	0000	LLI R0,00	2000	$R0 \leftarrow H'0000$
	0001	LLI R1,01	2101	$R1 \leftarrow H'0001$
	0002	IN R3, IP02	4302	$R3 \leftarrow IP02$ (dirección inicial)
	0003	IN R4, IP02	4402	$R4 \leftarrow IP02$ (dirección final)
	0004	ADDS RD, R3, R0	6D30	$RD \leftarrow R3$
	0005	LD R6, [00]	0600	$R6 \leftarrow M(pT)$ (valor mínimo inicial)
(a)	0006	ADDS R3,R3,R1	6331	$pT \leftarrow pT + 1$
	0007	ADDS RD, R3, R0	6D30	$r0 \leftarrow rD$
	0008	LD R5, [00]	0500	$dn \leftarrow M(pT)$
	0009	SUBS RF,R5,R6	7F56	$dn - vm$
	000A	LLI RD, 11	2D11	dirección de salto (b)
	000B	BS	C200	salta a (b) si $dn < vm$
(c)	000C	SUB RF,R3,R4	7F34	$pT - dF$
	000D	LLI RD, 14	2D14	dirección de salto (d)
	000E	BZ	C100	salta a (d) si $pT = dF$ (se ha llegado al final)
	000F	LLI RD, 06	2D06	dirección de salto (a)
	0010	BR	C000	Salto incondicional a (a)
(b)	0011	ADDS R6,R5,R0	6650	nuevo mínimo : $r6 \leftarrow r5$
	0012	LLI RD, 0C	2D0C	dirección de salto (c)
	0013	BR	C000	Salta a (c)
(d)	0014	OUT OP01,R6	5601	Salida del valor mínimo (vm)
	0015	HALT	F000	Fin

b) *Tiempo que tardaría en ejecutarse el programa*

Llamando N al número de datos que hay en la zona de memoria a considerar (df-di), en la siguiente tabla se indica el número de ciclos que consume cada instrucción

Nemónico	Nº de ciclos	Nº de veces que se ejecuta
LLI R0,00	6	1
LLI R1,01	6	1
IN R3, IP02	8	1
IN R4, IP02	8	1
ADDS RD, R3, R0	7	1
LD R6, [00]	9	1
ADDS R3,R3,R1	7	N-1
ADDS RD, R3, R0	7	N-1
LD R5, [00]	8	N-1
SUBS RF,R5,R6	7	N-1
LLI RD, 11	6	N-1
BS	6	N-1
SUBS RF,R3,R4	7	N-1

LLI RD, 14	6	N-1
BZ	6	N-1
LLI RD, 06	6	N-1
BR	6	N-1
ADDS R6,R5,R0	7	N-1
LLI RD, 0C	6	N-1
BR	6	N-1
OUT OP01,R6	8	1
HALT	6	1

Obsérvese que como el tiempo de ejecución del programa hay que calcularlo “en el peor de los casos” se ha supuesto que en todos los casos el valor del número es mínimo (es decir, en la zona de memoria están ordenados en forma descendiente).

Entonces el número total de ciclos consumidos en la ejecución del programa será:

$$NC = 44 + 91 \cdot (N-1) + 14 = 91 \cdot N + 149$$

Por otra parte, el periodo de reloj (tiempo de ciclo) es:

$$T = \frac{1}{F} = \frac{1}{10 \times 10^9} = 100 \mu s$$

con lo que el tiempo de ejecución del programa sería:

$$t = T \cdot NC = 100 \mu s \cdot (91N + 149) = (9,1 \cdot N + 14,9) ms$$

3. Suponga que en la posición de memoria A001 de CODE2 se encuentra la instrucción 47A3 (IN r7,IPA3). Rellenar la tabla siguiente, indicando las distintas microoperaciones que se implementarán al ejecutarse y cómo van cambiando los registros o posiciones de memoria que se muestran en las distintas columnas. Incluir tanto la fase de captación como de ejecución de la instrucción. Suponer que el usuario da en el puerto de entrada indicado el dato C3C3.

Solución:

		PC	AR	DR	IR	r7	M(A001)
Valores iniciales →		A001	70BC	A000	00BC	00AB	47A3
Fase	Microoperación						
Captación	AR ← PC	A001	A001	A000	00BC	00AB	47A3
	DR ← M(AR)	A001	A001	47A3	00BC	00AB	47A3
	IR ← DR	A001	A001	47A3	47A3	00AB	47A3
	PC ← PC+1	A002	A001	47A3	47A3	00AB	47A3
Ejecución	AR ← IR (0:7)	A002	00A3	47A3	47A3	00AB	47A3
	DR ← IPA3	A002	00A3	C3C3	47A3	00AB	47A3
	R7 ← DR	A002	00A3	C3C3	47A3	C3C3	47A3

Obsérvese que se ha tenido en cuenta que el código o dirección del puerto de entrada se encuentra en los 8 bits menos significativos del registro de instrucción; esto es en IR(7:0). En la fase de ejecución esta dirección se lleva al registro AR para que aparezca en el sub-bus de

direcciones del sistema, y así pueda el puerto (IPA3) dar al bus de datos su contenido, que es cargado en el registro DR, como se hace con todos los datos e información que se lleva al procesador. Por último, para completar la instrucción, se carga el registro r7 con el contenido del registro DR.

- 4.** En un computador se presentan cuatro procesos, cuyas prioridades, instantes de llegada y millones de instrucciones a ejecutar por el procesador se indican en la tabla. El procesador utilizado es de una frecuencia de reloj de $F = 1 \text{ GHz}$ y, por término medio, emplea dos ciclos en ejecutar cada instrucción. El sistema operativo invierte (5 ms) para realizar la planificación cuando concluye un proceso o cuando debe realizar cambios de contexto.

Proceso	Prioridad (1 la mayor)	Instante de llegada (ms)	Nº de instrucciones (millones)	Tiempo de CPU (ms)
P1	3	0	15	
P2	4	15	10	
P3	2	25	7,5	
P4	1	40	17,5	

- a) Obtener los tiempos de ejecución de cada proceso (incluirlos en la última columna de la tabla).**

El tiempo de procesador será el producto del nº de instrucciones (n) del proceso por lo que tarda en ejecutarse cada una de ellas. Y el tiempo de ejecución de cada instrucción se puede estimar como el producto del número de ciclos medio de cada instrucción (N_c) por el periodo de reloj ($T = 1/F$); es decir:

$$t = \frac{N_I \cdot N_c}{F}$$

En nuestro caso, $N_c = 2$ y $F = 1 \text{ GHz}$; es decir, se tiene que:

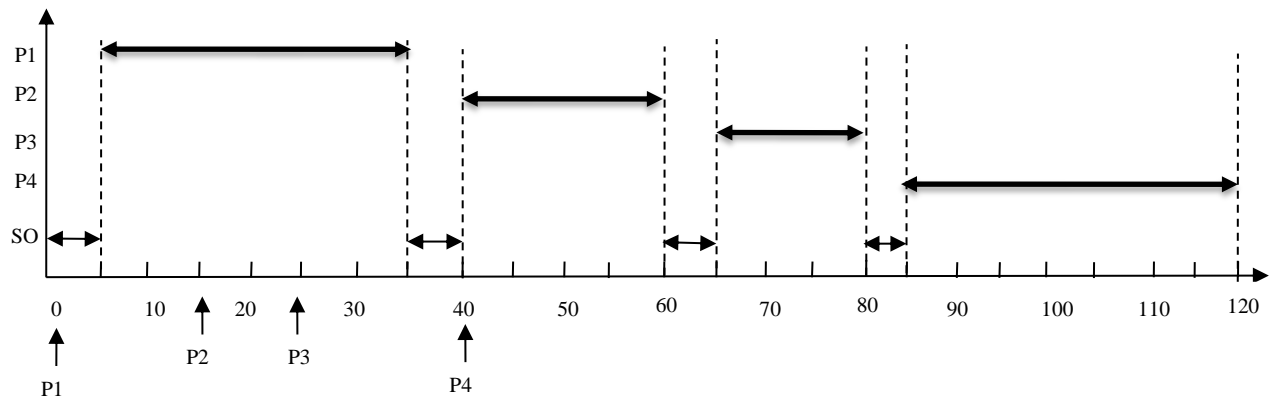
$$t = \frac{n \cdot 10^6 \cdot 2}{10^9} = 2 \cdot n \text{ milisegundos}$$

Es decir, tenemos los tiempos de procesador que se indican en la última columna de la siguiente tabla:

Proceso	Prioridad (1 la mayor)	Instante de llegada (ms)	Tiempo de CPU (ms)
P1	3	0	30
P2	4	15	20
P3	2	25	15
P4	1	40	35

- b) Obtener el coeficiente de respuesta del proceso P3 suponiendo planificación FCFS (First Come, First Served) y sistema operativo no apropiativo.**

La prioridad es $P1 \rightarrow P2 \rightarrow P3 \rightarrow P4$; es decir, cuando hay varios procesos pendientes de ejecutar, se ejecuta primero el que de ellos esté antes en la lista anterior.



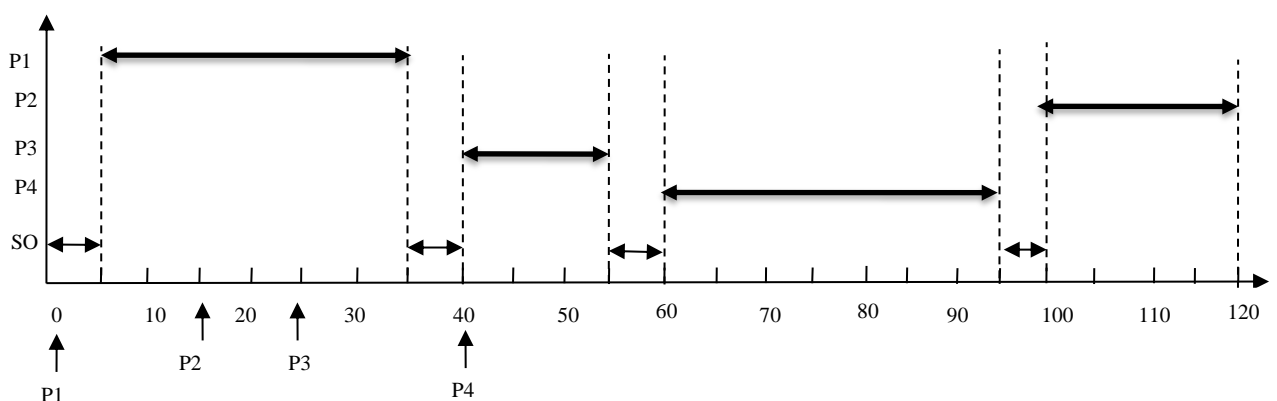
Por definición el tiempo de respuesta (t_r), es el tiempo comprendido entre el instante que llega el proceso (t_0) y el tiempo en que finaliza el mismo (t_f); y el coeficiente de respuesta (R) es el tiempo de respuesta dividido por el tiempo de procesamiento (t_{CPU}); es decir, para el proceso B se tiene:

$$R_{P3} = \frac{t_{r3}}{t_{CPU,3}} = \frac{t_{f3} - t_{03}}{t_{CPU,3}} = \frac{80 - 25}{15} = 3,67$$

Resultado	3,67
-----------	------

c) Obtener el coeficiente de respuesta del proceso P3 suponiendo planificación por prioridades y sistema operativo no apropiativo.

Ahora la prioridad es $P4 \rightarrow P3 \rightarrow P1 \rightarrow P2$; es decir, cuando hay varios procesos pendientes de ejecutar, se ejecuta primero el que de ellos esté antes en la lista anterior.

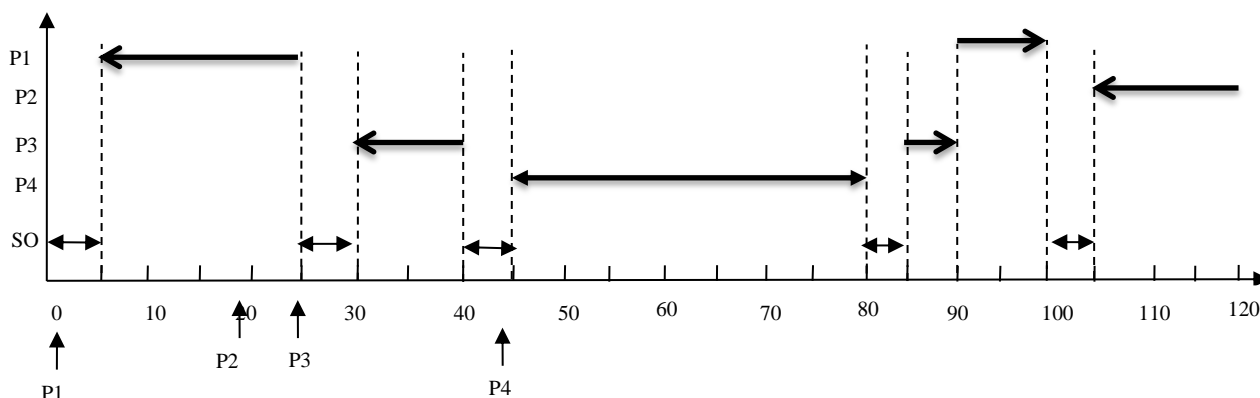


$$R_{P3} = \frac{t_{r3}}{t_{CPU,3}} = \frac{55 - 25}{15} = 2$$

Resultado	2
-----------	---

d) Obtener el coeficiente de respuesta del proceso P3 suponiendo planificación por prioridades y sistema operativo apropiativo (es decir, el SO interrumpe a un proceso en ejecución cuando llega otro más prioritario)

La prioridad sigue siendo P4 → P3 → P1 → P2; pero ahora en el momento que llegue un nuevo proceso con prioridad mayor que el que se esté ejecutando, el SO interrumpe a éste para dar paso al más prioritario.

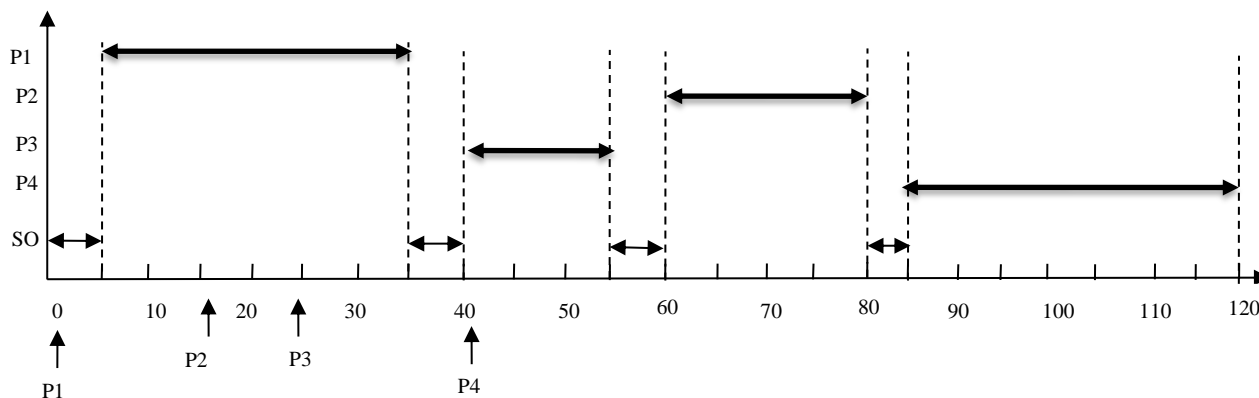


$$R_{P3} = \frac{t_{r3}}{t_{CPU,3}} = \frac{90 - 25}{15} = 4,3$$

Resultado	4,3
-----------	-----

e) Obtener el coeficiente de respuesta del proceso P3 suponiendo planificación SPN (*Shortest Process Next*) y sistema operativo no apropiativo.

Ahora la prioridad es P3 → P2 → P1 → P4; es decir, cuando hay varios procesos pendientes de ejecutar, se ejecuta primero el que de ellos esté antes en la lista anterior.



$$R_{P3} = \frac{t_{r3}}{t_{CPU,3}} = \frac{55 - 25}{15} = 2$$

Resultado	2
-----------	---