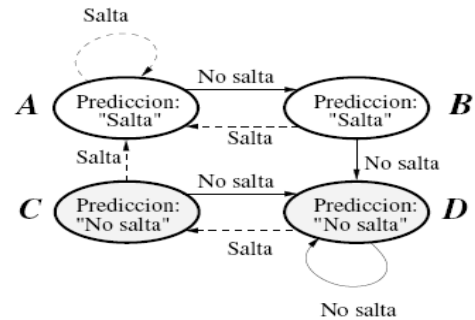


**Pregunta.** En la secuencia de instrucciones siguientes:

Addr	Inst	
0x03	addi r1,r0,#6	; r1 = 6
0x04	add r4,r0,r1	; r4 = r1
0x05	lw r2,dato	; r2 = dato
0x06	add r5,r2,r0	; r5 = dato
0x07 inicio:	subi r3,r1,r2	; r3 = r1 - r2
0x08	beqz r3, final	; si r3=0 saltar a <i>final</i>
0x09	addf f3,f2,f1	; f3 = f2 + f1
0x0A	beqz r5,final	; si r5=0 saltar a <i>final</i>
0x0B	addf f3,f2,f1	; f3 = f2 + f1
0x0C	addi r2,r2,#1	; r2 = r2 + 1
0x0D	subi r5,r5,#1	; r5 = r5 - 1
0x0E	subi r4,r4,#1	; r4 = r4 - 1
0x0F	bnez r4,inicio	; saltar a <i>inicio</i> si r4 es distinto de 0
....		
0x2C final:		

- a) Suponer que el procesador utiliza BTB de 4 entradas y 2bits de predicción. Realizar una traza de ejecución del código, mostrando el contenido de la BTB, (suponer dato=3 y BTB inicialmente vacía). Cuando se añade una nueva entrada en el BTB, su estado sería A si el salto ha sido efectivo y D si el salto no ha sido efectivo.



Dir salto	Dir destino	Bits predicción

- b) Si el código se ejecuta en un procesador segmentado con las siguientes etapas. Evaluar la penalización cuando un salto se predice mal.

**IF** Búsqueda de la instrucción.

**ID** Dec. de la instrucción y lectura de regs, lectura bits predicción y dir destino de salto

**ALU** Cálculo de la dirección de acceso a memoria.

**MEM** Acceso a memoria.

**EX1** Primera fase de ejecución y cálculo de la condición de salto.

**EX2** Segunda fase de ejecución.

**WB** Escritura en registros.

**Pregunta.** En la secuencia de instrucciones siguientes:

Addr	Inst	
0x03	addi r1,r0,#6	; r1 = 6
0x04	add r4,r0,r1	; r4 = r1
0x05	lw r2,dato	; r2 = dato
0x06	add r5,r2,r0	; r5 = dato
0x07 inicio:	subi r3,r1,r2	; r3 = r1 - r2
0x08	beqz r3, final	; si r3=0 saltar a <i>final</i>
0x09	addf f3,f2,f1	; f3 = f2 + f1
0x0A	beqz r5,final	; si r5=0 saltar a <i>final</i>
0x0B	addf f3,f2,f1	; f3 = f2 + f1
0x0C	addi r2,r2,#1	; r2 = r2 + 1
0x0D	subi r5,r5,#1	; r5 = r5 - 1
0x0E	subi r4,r4,#1	; r4 = r4 - 1
0x0F	bnez r4,inicio	; saltar a <i>inicio</i> si r4 es distinto de 0
....		
0x2C final:		

- a) Calcular la penalización efectiva debida a los saltos (siendo dato=3) en los siguientes casos
- a1)(0,5) predicción dinámica de 1 bit
- a2) (1)predicción dinámica de 3 bits

**Nota: :** La penalización por saltos incorrectamente predichos es de 4 ciclos y para los saltos correctamente predichos es 0 ciclos. Para hacer la predicción en la primera ejecución de las instrucciones de salto se utiliza el siguiente esquema de predicción estática:

	1bit	3bits
Hacia adelante	0	000
Hacia atrás	1	111

- b) Suponer que el procesador utiliza BTB de 4 entradas con **predicción implícita**. Realizar una traza de ejecución del código, mostrando el contenido de la BTB, (suponer dato=3 y BTB inicialmente vacía).