

Examen de Arquitectura de Computadoras

20 de febrero de 2020

Instrucciones:

- **Indique su nombre, apellido y número de cédula en todas las hojas que entregue.**
- **Escriba las hojas de un solo lado.**
- **Las hojas deben estar numeradas y en la primer hoja debe escribirse el total.**
- **Empiece cada ejercicio en una hoja nueva.**
- **No puede utilizar material ni calculadora.**
- **Apague su celular.**
- **La duración del examen es de tres horas. En dicho tiempo debe también completar sus datos. Solo se contestarán dudas de letra. No se aceptan preguntas en los últimos 30 minutos del examen.**
- **Para aprobar debe contestar correctamente un ejercicio entero y dos preguntas.**

Pregunta 1

Dada una CPU de 8 bits y 4 registros A, B, C, D; que cuenta con el siguiente conjunto de instrucciones:

Operación	Descripción (Ri es cualquiera de los 4 registros A, B, C, D.)
NOP	No operación
LD R1, R2	Carga en registro R1, el contenido de la dirección de memoria contenida en registro R2
ST R1, R2	Carga en la dirección de memoria contenida en registro R2 , el contenido del registro R1
MVI I,K	Carga en el registro A el valor inmediato I de 4 bits. Si K=0, I se carga en la parte baja de A, sino se carga en la parte alta.
MVR R1, R2	Carga el contenido del registro R1 en el registro R2
NOT R1	Realiza el complemento bit a bit del contenido del registro R1
AND R1, R2	Realiza el AND bit a bit del contenido de los registros R1 y R2 y lo acumula en A
ADD R1, R2	Realiza la suma del contenido de los registros R1 y R2 y lo acumula en A

Diseñe un formato de instrucción y defina los códigos de operación correspondientes para una máquina RISC de 8 bits con el conjunto de instrucciones planteado.

Pregunta 2

a) Indique qué modos de direccionamiento son soportados por la arquitectura 8086.

b) En el siguiente fragmento de código, identifique los modos de direccionamiento de los operandos en cada una de las instrucciones:

```
MOV AX, [1234]
ADD AX, 1
SUB BX, AX
MOV ES:[BX], DX
CMP DX, [SI + 0x400]
```

Pregunta 3

Multiplique los números, cuyas representaciones en punto flotante media precisión son:

$$0110010100000101 \times 1101001001100000$$

Pregunta 4

Defina brevemente: Hit Time, Miss Rate, Miss Penalty y Tiempo Promedio de Acceso a Memoria.

Escriba la ecuación para el cálculo del tiempo promedio de acceso a memoria en un esquema con dos niveles de memoria caché.

Problema 1

Se requiere diseñar un controlador de interrupciones para una CPU que cuenta con una entrada de solicitud de interrupción (*interrupt request*, INTR, que es salida del controlador) y una salida de aceptación de interrupción (*interrupt acknowledge*, INTA, que es entrada del controlador). El controlador debe ser capaz de atender interrupciones generadas por dos fuentes externas (señales IRQ1 e IRQ0, que son entradas al controlador) y generar las correspondientes IA1 e IA0 en función de cuál pedido debe aceptar. Todas las señales están sincronizadas con una fuente de reloj (señal CLK).

El funcionamiento del controlador se indica a continuación;

El controlador debe mantener su salida INTR en 0 mientras espera pedidos de interrupción. Al recibir un pedido de interrupción por las señales IRQ1 y/o IRQ0, el controlador debe poner su salida INTR en 1 hasta que la CPU coloque un 1 en la señal INTA. Cuando se activa la INTA, el controlador escribe un 1 en su salida IA1 o en la IA0, durante un ciclo de reloj, dependiendo de cual de los dos pedidos ha llegado primero, en caso que al momento de activarse la señal INTA las dos entradas IRQ estuvieran en 1. Si las dos entradas IRQ pasaran a 1 al mismo tiempo, se debe atender primero a IRQ1.

Se asume que el dispositivo que solicita la interrupción pasa a 0 la señal IRQ inmediatamente luego de que se activa la señal IA correspondiente.

Se solicita:

Implementar el controlador de interrupciones descrito, utilizando la metodología del curso. Se dispone de flip-flops tipo D y compuertas lógicas.

Problema 2

La empresa **LetItFlow** le ha encargado el desarrollo de un sistema de control de las compuertas de una represa hidroeléctrica conectada a la red de UTE.

La represa consiste en un embalse que retiene el agua del río y una compuerta reguladora del caudal de agua, conectada a una turbina hidráulica encargada de generar la energía eléctrica en base a dicho caudal (mayor flujo de agua equivale a mayor potencia generada). La potencia máxima de generación de la turbina es de 50 MW.

Debido a que se dispone en el país de otros medios para generar energía eléctrica, y que además la demanda de energía es variable a lo largo del día, existe un organismo (el Despacho Nacional de Carga = DNC) encargado de regular la potencia generada por la represa en cada instante.

El sistema de control debe funcionar de la siguiente manera:

- Cada vez que el DNC solicite una nueva potencia, se debe abrir o cerrar la compuerta para adecuar la potencia generada por la turbina.
- Una vez que dicha potencia es alcanzada, se debe detener el motor que controla la compuerta.
- Para cambiar el sentido de giro del motor, primero se debe detener el motor y luego encenderlo con el nuevo sentido.
- Cada vez que el motor es detenido se debe esperar 5 minutos antes de volver a activarlo.
- En caso de emergencia se debe proceder a cerrar totalmente la compuerta y esperar 30 minutos antes de que el sistema pueda volver a estar disponible.

El sistema consta de los siguientes elementos:

- El DNC genera una interrupción cada vez que desea cambiar la potencia a generar por la represa, invocando a la rutina **potencia_solicitada()**. El valor de la nueva potencia solicitada queda accesible en los 6 bits menos significativos del puerto de E/S de solo lectura en la dirección **POTENCIA** (el contenido del puerto se pierde una vez leído).
- Un timer que interrumpe con una frecuencia de 10 Hz invocando a la rutina **tiempo()**.
- Una rutina **emergencia()** que es invocada en caso de emergencia.
- Una rutina **tope()** que es invocada cada vez que la compuerta está totalmente cerrada o totalmente abierta.
- Un sensor que mide la potencia actual generada por la turbina. Este sensor es accesible en los 6 bits menos significativos del puerto de E/S (byte de solo lectura) en la dirección **TURBINA**.
- Puerto de 8 bits de solo escritura en la dirección **COMPUERTA**, en cuyo bit menos significativo se activa el motor (1 = activado) y en el bit b1 se indica el modo (1 = apertura, 0 = cierre).

Se pide:

Escribir en un lenguaje de alto nivel (preferentemente C) todas las rutinas necesarias para implementar el sistema de control descrito.

Nota: se puede asumir que nunca se solicitará mayor potencia que la que puede generar la represa.

Respuesta Pregunta 1

En la siguiente tabla se presenta el código de operación asignado a cada instrucción y el formato de instrucción para cada caso.

Opr	Código	Formato
NOP	000	OP ₂ OP ₁ OP ₀ X X X X X
LD	001	OP ₂ OP ₁ OP ₀ R ₁ R ₁ ₀ R ₂ R ₂ ₀ X
ST	010	OP ₂ OP ₁ OP ₀ R ₁ R ₁ ₀ R ₂ R ₂ ₀ X
MVI	011	OP ₂ OP ₁ OP ₀ I ₃ I ₂ I ₁ I ₀ K
MVR	100	OP ₂ OP ₁ OP ₀ R ₁ R ₁ ₀ R ₂ R ₂ ₀ X
NOT	101	OP ₂ OP ₁ OP ₀ R ₁ R ₁ ₀ X X X
AND	110	OP ₂ OP ₁ OP ₀ R ₁ R ₁ ₀ R ₂ R ₂ ₀ X
ADD	111	OP ₂ OP ₁ OP ₀ R ₁ R ₁ ₀ R ₂ R ₂ ₀ X

Respuesta Pregunta 2

a) Los modos de direccionamiento soportados por 8086 son:

- inmediato
- directo a registro
- directo a memoria
- indirecto por registro
- indizado (combinación de indirecto por registro con directo a memoria)

b)

- MOV AX, [1234] → directo a registro, directo a memoria
- ADD AX, 1 → directo a registro, inmediato
- SUB BX, AX → directo a registro, directo a registro
- MOV ES:[BX], DX → indirecto por registro, directo a registro
- CMP DX, [SI + 0x400] → directo a registro, indizado

Respuesta Pregunta 3

$$0 \ 11001 \ 0100000101 = 1,0100000101 \times 2^{10}$$

$$1 \ 10100 \ 1001100000 = -1,10011 \times 2^5$$

signo: + * - = -

sumo exponentes: $10+5 = 15$

multiplico mantisas:

$$\begin{array}{r} 1,0100000101 \\ \times 1,10011 \\ \hline 10100000101 \\ 10100000101 \\ 10100000101 \\ 10100000101 \\ \hline 1,111111111111111 \end{array}$$

$$\Rightarrow -1,111111111111111 \times 2^{15} = -65535$$

Respuesta Pregunta 4

Hit Time: Tiempo de acceso promedio en el nivel de jerarquía considerado (donde se da el hit).

Miss Rate: Es la tasa de fallos en encontrar un elemento de información en el lugar buscado (y coincide con $1 - \text{Hit Rate}$).

Miss Penalty: Tiempo de acceso promedio adicional requerido para acceder al elemento de información en el nivel de jerarquía inferior. Típicamente ocurre que $\text{Hit Time} \ll \text{Miss Penalty}$.

Tiempo promedio de acceso a memoria: Si consideramos el nivel de jerarquía memoria caché podemos establecer que el tiempo promedio de acceso a memoria (es decir a elementos de información almacenados en la memoria principal de la computadora) cumple la relación: $\text{Tiempo promedio de acceso a memoria} = \text{Hit Time} + \text{Miss Rate} * \text{Miss Penalty}$

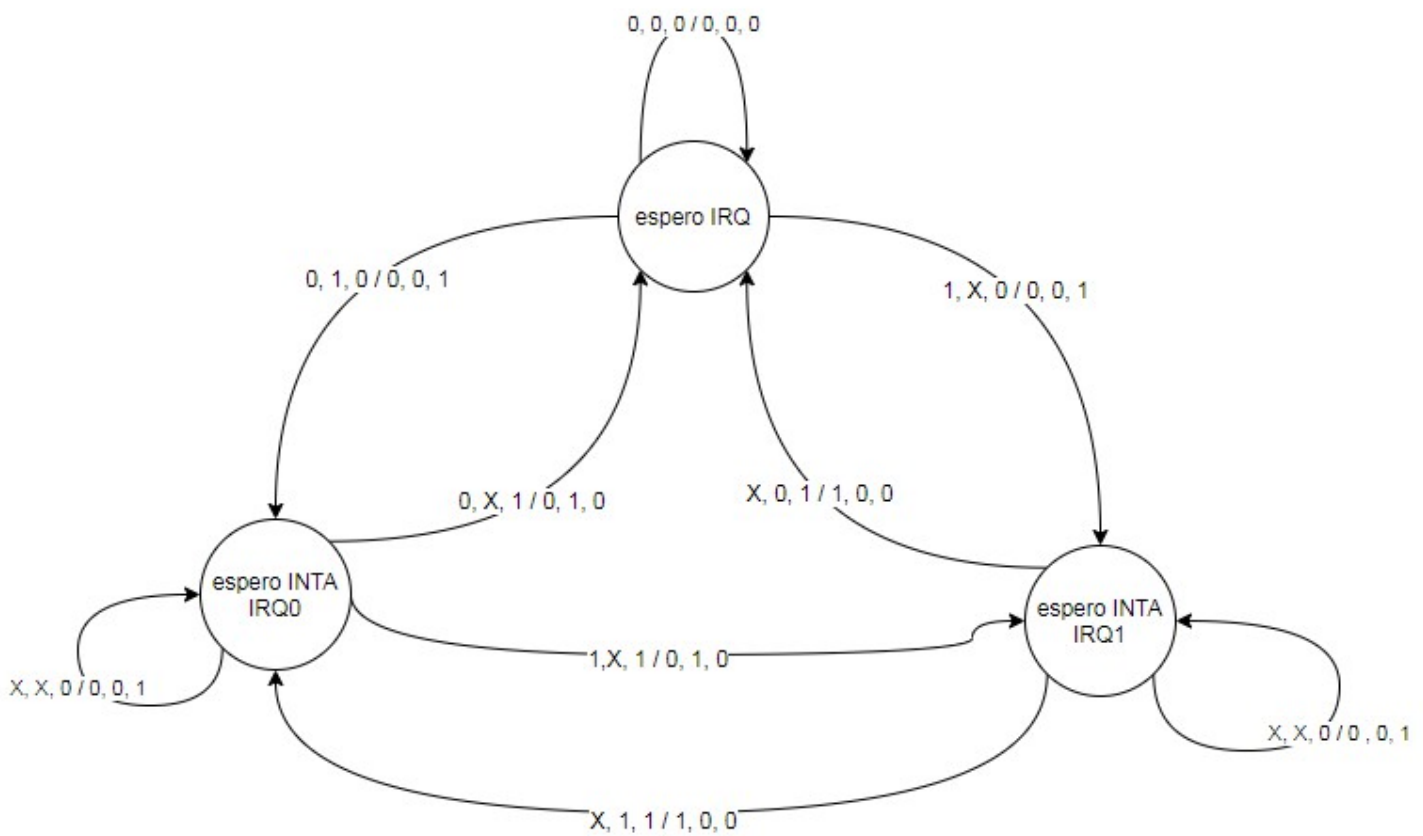
La ecuación de tiempo promedio de acceso a memoria en un esquema con dos niveles de memoria caché es la siguiente:

$$\text{TPAM} = \text{HitTime}_{L1} + \text{MissRate}_{L1} * (\text{HitTime}_{L2} + \text{MissRate}_{L2} * \text{MissPenalty}_{\text{RAM}})$$

En dónde los subíndices L1, L2 identifican a los datos para los niveles de caché L1 y L2, mientras que RAM refiere a los datos para la memoria RAM.

Solución Problema 1**Diagrama de estados**

IRQ1, IRQ0, INTA / IA1, IA0, INTR

**Codificación de estados:**

Con 3 estados, se precisan dos flip flops para codificar los estados.

'espero IRQ' → 10

'espero inta IRQ0' → 00

'espero inta IRQ1' → 01

Tabla de verdad:

Estado actual (q1, q0)		Entrada (IRQ1, IRQ0, INTA)			Próximo estado (d1, d0)		Salida (IA1, IA0, INTR)		
0	0	0	0	0	0	0	0	0	1
0	0	0	0	1	1	0	0	1	0
0	0	0	1	0	0	0	0	0	1
0	0	0	1	1	1	0	0	1	0
0	0	1	0	0	0	0	0	0	1
0	0	1	0	1	0	1	0	1	0
0	0	1	1	0	0	0	0	0	1
0	0	1	1	1	0	1	0	1	0
0	1	0	0	0	0	0	0	0	1
0	1	0	0	1	1	0	1	0	0
0	1	0	1	0	0	0	0	0	1
0	1	0	1	1	0	0	1	0	0
0	1	1	0	0	0	0	0	0	1
0	1	1	0	1	1	0	1	0	0
0	1	1	1	0	0	0	0	0	1
0	1	1	1	1	0	0	1	0	0
1	0	0	0	0	1	0	0	0	0
1	0	0	0	1	X	X	X	X	X
1	0	0	1	0	0	0	0	0	1
1	0	0	1	1	X	X	X	X	X
1	0	1	0	0	0	1	0	0	1
1	0	1	0	1	X	X	X	X	X
1	0	1	1	0	0	1	0	0	1
1	0	1	1	1	X	X	X	X	X
1	1	0	0	0	X	X	X	X	X
1	1	0	0	1	X	X	X	X	X
1	1	0	1	0	X	X	X	X	X
1	1	0	1	1	X	X	X	X	X
1	1	1	0	0	X	X	X	X	X
1	1	1	0	1	X	X	X	X	X
1	1	1	1	0	X	X	X	X	X
1	1	1	1	1	X	X	X	X	X
1	1	1	1	1	X	X	X	X	X

Diagramas de Karnaugh

d1:

q1 = 0

q0,irq1/irq0, inta	00	01	11	10
00	0	1	1	0
01	0	0	0	0
11	0	1	1	0
10	0	0	0	0

q1 = 1

q0,irq1/irq0, inta	00	01	11	10
00	1	X	X	0
01	0	X	X	0
11	X	X	X	X
10	X	X	X	X

$$d1 = !Q0 \cdot !IRQ1 \cdot INTA + Q0 \cdot IRQ1 \cdot INTA + !IRQ0 \cdot !IRQ1 \cdot Q1$$

d0:

q1 = 0

q0,irq1/irq0, inta	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	0	0	0
10	0	0	0	0

q1 = 1

q0,irq1/irq0, inta	00	01	11	10
00	0	X	X	0
01	1	X	X	1
11	X	X	X	X
10	X	X	X	X

$$d0 = \neg IRQ1 \cdot INTA \cdot \neg IQ0 + IRQ1 \cdot Q1$$

IA1:

q1 = 0

q0,irq1/irq0, inta	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	1	1	0
10	0	1	1	0

q1 = 1

q0,irq1/irq0, inta	00	01	11	10
00	0	X	X	0
01	0	X	X	0
11	X	X	X	X
10	X	X	X	X

$$IA1 = Q0 \cdot INTA$$

IA0:

q1 = 0

q0,irq1/irq0, inta	00	01	11	10
00	0	1	1	0
01	0	1	1	0
11	0	0	0	0
10	0	0	0	0

q1 = 1

q0,irq1/irq0, inta	00	01	11	10
00	0	X	X	0
01	0	X	X	0
11	X	X	X	X
10	X	X	X	X

$$\text{IA0} = \text{!Q0} \cdot \text{INTA}$$

INTR:

q1 = 0

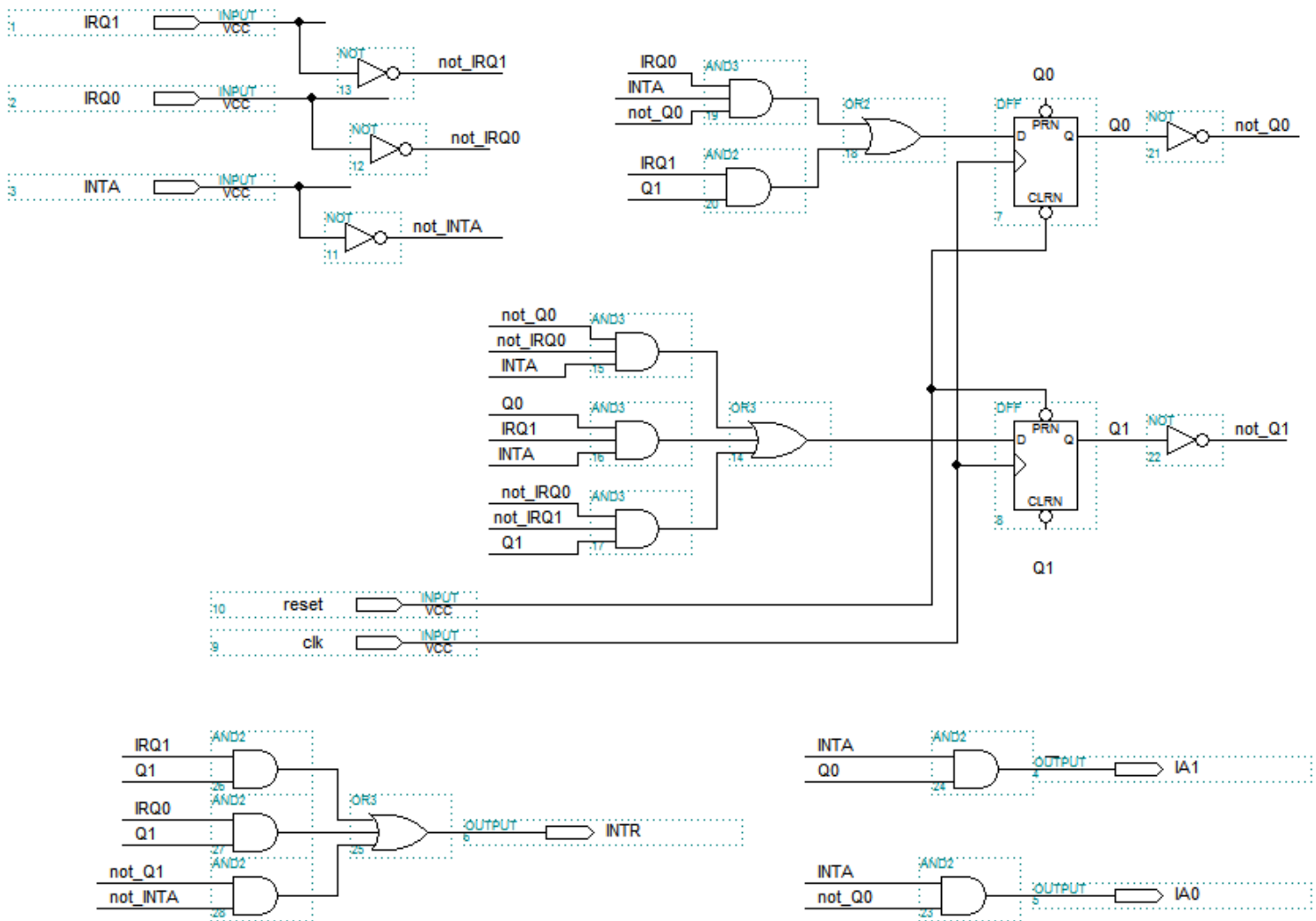
q0,irq1/irq0, inta	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	0	0	1
10	1	0	0	1

q1 = 1

q0,irq1/irq0, inta	00	01	11	10
00	0	X	X	1
01	1	X	X	1
11	X	X	X	X
10	X	X	X	X

$$\text{INTR} = \text{IRQ1} \cdot \text{Q1} + \text{IRQ0} \cdot \text{Q1} + \text{!Q1} \cdot \text{!INTA}$$

Circuito



Solución Problema 2

```
#define CERRAR 1
#define ABRIR 3
#define APAGAR 0
#define T_5_MIN 3000
#define T_30_MIN 18000
#define ESPERO 0
#define ABRIENDO 1
#define CERRANDO 2
#define EMERGENCIA_CERRAR 3

void interrupt potencia_solicitada()
{
    objetivo = in(POTENCIA) & 0x3F;
    int sensor = in(TURBINA) & 0x3F;
    char apagar = 0;
    if(objetivo > sensor)
    {
        if(estado == CERRANDO) apagar = 1;
        estado = ABRIENDO;
    }
    else if(objetivo < sensor)
    {
        if(estado == ABRIENDO) apagar = 1;
        estado = CERRANDO;
    }
    else estado = ESPERO;
    if(apagar && tics == 0)
    {
        out(COMPUERTA, APAGAR);
        tics = T_5_MIN;
    }
}

void interrupt tiempo()
{
    if(tics > 0)
        tics--;
}

void interrupt emergencia()
{
    estado = EMERGENCIA_CERRAR;
}

void interrupt tope()
{
    if(estado != ABRIENDO)
        cerrada = true;
}

int estado;
int tics;
char cerrada;
int objetivo;
```

```
void main()
{
    // instalar ISRs
    tics = 0;
    estado = ESPERO;
    cerrada = false;
    enable();
    while(true)
    {
        switch(estado)
        {
            case ESPERO: break;
            case ABRIENDO:
                if(tics == 0)
                {
                    if(in(TURBINA) & 0x3F >= objetivo)
                    {
                        out(COMPUERTA, APAGAR);
                        tics = T_5_MIN;
                        estado = ESPERO;
                    }else
                    {
                        out(COMPUERTA, ABRIR);
                        cerrada = false;
                    }
                }
                break;
            case CERRANDO:
                if(tics == 0)
                {
                    if(in(TURBINA) & 0x3F <= objetivo || cerrada)
                    {
                        out(COMPUERTA, APAGAR);
                        tics = T_5_MIN;
                        estado = ESPERO;
                    }else out(COMPUERTA, CERRAR);
                }
                break;
            case EMERGENCIA_CERRAR:
                if(tics == 0 || cerrada)
                {
                    if(!cerrada)
                    {
                        cerrada = false;
                        while(!cerrada) out(COMPUERTA, CERRAR);
                    }
                    out(COMPUERTA, APAGAR);
                    tics = T_30_MIN;
                    estado = espero;
                    while(tics > 0) {}
                }
                break;
        }
    }
}
```