

ET1032 Informática Industrial

8 de julio de 2015

Soluciones al examen.

NOTA: El nivel de detalle con que se responde a las preguntas en este documento no es el que se requiere ni se espera para la calificación de los ejercicios. Al redactar este documento se pretende dar información más amplia y completa con fines documentales y didácticos.

Pregunta 1. (1 punto) – Indicad qué dos tipos fundamentales de información se encuentran, convenientemente codificados según su formato, en las instrucciones máquina de los procesadores.

Una instrucción máquina es un conjunto de bits que reside en memoria con información codificada que el procesador es capaz de leer, de interpretar y de ejecutar llevando a cabo las operaciones que la propia instrucción indica, sobre los datos que también se indican en ella. De esta manera, una instrucción, contiene junto con otros posibles modificadores, información acerca de:

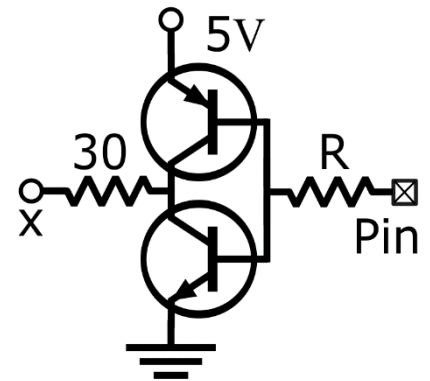
- La operación a llevar a cabo por el procesador, que se conoce generalmente como **código de operación**.
- La referencia a los operandos, datos de entrada y resultado, con que trabaja la instrucción. Más específicamente se indica el recurso del sistema donde obtener los datos y en qué almacenamiento dejar el resultado, mediante los correspondientes **modos de direccionamiento**.

Pregunta 2. (1 punto) – Supóngase un procesador que dispone de una única línea de interrupción y, por tanto, de un solo vector, asociado a ella. Indicad justificadamente de qué manera se podrían, en este caso, establecer distintas prioridades entre los dispositivos capaces de generar interrupciones.

Un procesador con una sola línea de interrupción y con un solo vector para indicar la dirección de comienzo de la rutina de tratamiento de interrupciones **RTI** no puede discernir mediante su circuitería entre varias fuentes de interrupción y por tanto no puede disponer de ningún mecanismo *hardware* para establecer las prioridades entre los diferentes dispositivos conectados a tal línea. En este caso, como cualquier causa de interrupción dirige la ejecución a la misma **RTI**, es su propio código el que debe establecer las prioridades según el orden en que se realice la prueba de estado para ver cuáles de las posibles fuentes han generado la interrupción.

Aunque se podrían concebir y programar mecanismos complejos con prioridades variables o dinámicas, en el caso más simple el mero orden en que se consultan los bits de estado –los *flags* de interrupción– de los dispositivos habilitados para interrumpir al procesador es el que establece la prioridad entre ellos.

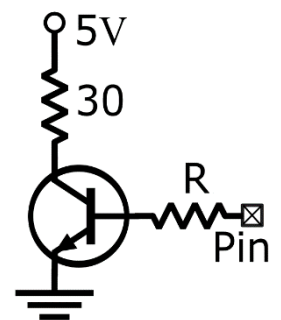
Pregunta 4. (1,5 puntos) – Sea el montaje de la figura, en que el punto **X** puede estar conectado a **5V** o a tierra (**0V**). Indicad qué transistor conducirá y a qué tensión debe estar conectado **X** para que circule corriente a través de la resistencia de 30Ω cuando el pin está a nivel alto. Análogamente, indicad lo mismo para que la resistencia conduzca corriente con el pin a nivel bajo. Calculad un único valor de **R** para que el transistor que conduzca en cada caso esté saturado en todo el rango de tensiones del nivel, e indicad la corriente máxima y mínima de base y la que atraviesa la resistencia de 30Ω en cada caso. Las características de los transistores, iguales para ambos, y los valores de continua del microcontrolador, alimentado a 5V, aparecen en las tablas inferiores.



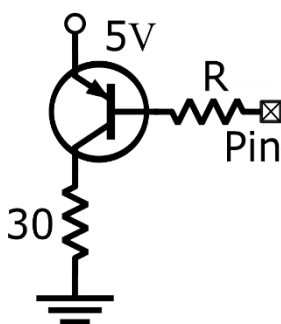
HFE _{min}	330	I _{Cmax}	200 mA	V _{CEsat}	200 mV	V _{BEsat}	650 mV
--------------------	-----	-------------------	--------	--------------------	--------	--------------------	--------

V _{IHmin}	3,6 V	V _{OHmin}	4,2 V	I _{Imax}	4 μA
V _{ILmax}	1,2 V	V _{OLmax}	0,6 V	I _{Omax}	23 mA

El circuito mostrado es una configuración típica de las salidas cuando se desea aumentar la corriente suministrada por el microcontrolador mediante los transistores. Como se ve, cuando el pin está a **nivel alto** el transistor **PNP** superior no puede conducir, no solo porque la tensión en el pin no permite polarizar correctamente su diodo emisor-base sino porque el pin no permite que circule corriente hacia su interior. En este caso el transistor **NPN** inferior sí está correctamente polarizado y puede conducir sin problemas. Para ello, es necesario que la tensión de colector sea superior a la de emisor, por lo tanto el punto **X** debe estar conectado a **5V**. Dado que el transistor **PNP** no conduce, el **NPN** sí y **X** está conectado a tierra podemos analizar el circuito como el equivalente de la figura de la derecha.



Circuito equivalente con el pin a nivel alto



Circuito equivalente con el pin a nivel bajo

Por otra parte con un **nivel bajo** en el pin es el transistor **NPN** el que no puede conducir por las mismas razones –el pin no suministra corriente– mientras que el **PNP** sí conducirá siempre que la tensión en su colector sea inferior a la del emisor, lo que se verifica cuando conectamos el punto **X** a tierra. En este caso, el circuito equivalente es el que aparece a la izquierda de este texto.

En ambos circuitos equivalentes y dado que las características de los transistores son idénticas, la corriente de colector con el transistor saturado será la misma, y vendrá dada por la expresión

$$I_c = \frac{5 - V_{CEsat}}{30 \Omega} = \frac{4,8 V}{30 \Omega} = 160 mA$$

Así pues, basta con calcular **R** para que en el peor de los casos el transistor correspondiente esté saturado, y esto garantizará que se satura en todos los demás. Este peor caso se da cuando la diferencia de tensión entre el emisor del transistor y el pin es menor, lo que ocurre con el primer circuito, dado que esa diferencia es **V_{OHmin}**, es decir **4,2V**.

Así pues elegimos **R** para que en este caso la intensidad de base sea diez veces mayor que la intensidad de saturación mínima, según

$$I_B = 10 * \frac{I_C}{h_{FE}} = \frac{1600 \text{ mA}}{330} = 4,85 \text{ mA}$$

y de esta manera calculamos **R** con la siguiente igualdad

$$I_B = 4,85 \text{ mA} = \frac{V_{Pin} - V_{BEsat}}{R} = \frac{4,2 \text{ V} - 0,65 \text{ V}}{R} \Rightarrow R = 732 \Omega$$

Una vez obtenido el valor de **R** y sabiendo que la intensidad de colector será siempre de 160 mA, nos queda por calcular las corrientes máximas y mínimas que atraviesan R en cada caso. Aplicando la formula anterior cuando el emisor está conectado a tierra –pin a nivel alto- y cambiando el numerador por $5V - V_{BEsat} - V_{Pin}$ cuando está a nivel bajo, se obtienen los valores pedidos, que se presentan en la siguiente tabla:

Pin a nivel alto		Pin a nivel bajo	
V _{pin}	I _b	V _{pin}	I _b
5 V	5,94 mA	0V	5,94 mA
4,2 V	4,85 mA	0,6V	5,12 mA

Pregunta 5. (2,5 puntos) – Dado el montaje de la figura y el código que aparece en el recuadro, y considerando las características del microcontrolador que aparecen en la tabla anterior y **1,8V** de caída de tensión en el diodo LED, indicad:

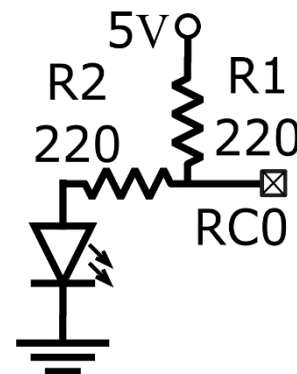
- Las corrientes máximas y mínimas que atraviesan cada una de las resistencias y el diodo, en los instantes **t1**, **t2** y **t3**, es decir, después de haber ejecutado las instrucciones hasta el comentario correspondiente.
- Los valores (**1** o **0**) de las variables **v1**, **v2** y **v3** al terminar la ejecución del fragmento de código.

Justificad adecuadamente todas las respuestas.

```

unsigned char v1, v2, v3;

TRISCbits.TRISC0 = 1;
v1 = PORTCbits.PORTC0;    // t1
...
LATCbits.LATC0 = 0;
TRISCbits.TRISC0 = 0;
v2 = PORTCbits.PORTC0;    // t2
...
LATCbits.LATC0 = 1;
v3 = PORTCbits.PORTC0;    // t3
    
```



Los parámetros eléctricos del circuito variarán significativamente en función de la configuración del pin y su nivel lógico, lo que se modifica en tres etapas al ejecutar el programa adjunto. En primer lugar, cuando se llega al instante etiquetado como **t1**, el pin está configurado como entrada –su bit del registro **TRIS** asociado está a **1**– así que en una primera aproximación consideraremos que no influye en el circuito. En este momento la corriente que atraviesa las dos resistencias y el LED viene dada por

$$I = \frac{5 \text{ V} - V_{LED}}{R1 + R2} = \frac{3,2 \text{ V}}{440 \Omega} = 7,27 \text{ mA}$$

En este caso, como se ve, la corriente máxima de entrada en el pin es unas 2000 veces menor, por lo que podemos despreciarla sin problemas. Si pese a ello queremos considerarla, la nueva expresión para calcular la intensidad que atraviesa **R1** vendría dada por la ley de Ohm según la ecuación

$$5 V = I \cdot R1 + (I - 4 \mu A) \cdot R2 + 1,8 V \Rightarrow I = 7,27 mA$$

que como era de esperar es igual en sus dos primeras cifras decimales a la calculada despreciando la intensidad de entrada. Así pues en este primer momento **t1** la intensidad que atraviesa **R1**, **R2** y el diodo LED es de **7,27mA**. En este caso la tensión en el pin será $5V - I \times R1$, es decir **3,4V**. Como este valor se encuentra entre **VIHmin** y **VILmax** no podemos saber si el microcontrolador lo entenderá como un **1** o un **0** lógicos, con lo que el valor de **v1** no se puede determinar.

De este instante al **t2** el pin ha sido configurado como salida y previamente se ha puesto a 0, así que la configuración eléctrica del circuito cambia totalmente. Si la tensión en el pin es la propia del nivel bajo, la corriente que atraviesa **R1** será absorbida totalmente por el pin, dado que el diodo no conducirá. Para asegurarnos hay que verificar que en el caso de máxima corriente esta no supere el valor **IOmax**, y para ello evaluamos

$$I = \frac{5 V - 0 V}{R1} = \frac{5 V}{220 \Omega} = 22,73 mA$$

Como vemos, no se supera el valor máximo de corriente aceptada por el pin con lo que podemos asegurar que la tensión en él será siempre menor o igual que **IOlmax**, es decir **0,6V**. Así pues en el instante **t2** la corriente que atraviesa el diodo y **R2** será **0** dado que el diodo no puede conducir, la corriente máxima que atraviesa **R1** será la indicada, **22,73mA** y la mínima vendrá dada por

$$I = \frac{5 V - 0,6 V}{R1} = \frac{4,4 V}{220 \Omega} = 20 mA$$

El valor de **v2** será **0** dado que leemos el valor del pin que siempre será inferior a **0,6V** como hemos dicho.

En el instante **t3** el valor del pin ha cambiado de nivel bajo a alto. Como hemos hecho antes, comprobamos en primer lugar que no se supere la corriente máxima, ahora con la expresión

$$I = \frac{5 V - 1,8 V}{R2} = \frac{3,2 V}{220 \Omega} = 14,55 mA$$

y vemos nuevamente que la corriente es adecuada. Podemos decir además que en este caso la intensidad que atraviesa el diodo y **R2** será máxima e igual al valor calculado arriba, mientras que la que atraviesa **R1** será mínima e igual a **0**. En el otro extremo de la tensión de salida del nivel alto calculamos la intensidad que atraviesa el diodo y **R2** mediante

$$I = \frac{4,2 V - 1,8 V}{R2} = \frac{2,4 V}{220 \Omega} = 10,91 mA$$

que es mínima, mientras que la que atraviesa **R1** es máxima y viene dada por

$$I = \frac{5 V - 4,2 V}{R1} = \frac{0,8 V}{220 \Omega} = 3,64 mA$$

Por las mismas razones que antes, el valor de **v3**, leído en este momento será **1**.

Pregunta 6. (2,5 puntos) – Realizad una función en C `int mezcla(int mascara, int dat0, int dat1)` que devuelva un entero formado con los bits de `dat0` en las posiciones en que los bits de la máscara están a 0, y los de `dat1` cuando los de la máscara están a 1.

Por ejemplo, la llamada `mezcla(0x0FF00F0F, 0xAAAAAAAA, 0x55555555)` devolvería `0xA55AA5A5`.

Utilizad esta función para resolver el ejercicio 5 del examen de junio, cuyo enunciado se copia:

Se quiere modificar dos campos, uno de 7 y otro de 5 bits, de un mismo registro de control de 32 bits. El primer campo comienza en el bit 11 y el segundo en el 24 –recordad que los bits se numeran a partir del menos significativo, cuyo índice es 0-. Se tienen dos enteros, `val1` y `val2` que contienen en sus bits menos significativos los valores a poner en dichos campos. Realizad un fragmento de código en lenguaje C que modifique los campos dados poniendo en ellos tales valores y dejando sin modificar el resto de bits del registro.

El código de la función pedida es sencillo. Para quedarnos con los bits correspondientes del primer valor haremos un **and** lógico entre la máscara negada –cambiando sus 0 por 1 y viceversa- y el valor `dat0`. Para quedarnos con los de `dat1` haremos de nuevo una operación **and** entre la máscara original y este valor. Posteriormente compondremos todos los bits en el resultado haciendo un **or** lógico entre ambos resultados. Una posible implementación de esto sería

```
int mezcla(int mascara, int dat0, int dat1)
{
    dat0 = dat0 & ~mascara; // Seleccionamos los bits de dat0, 0 en la mascara
    dat1 = dat1 & mascara;  // Luego los de dat1, 1 en la mascara
    return dat0 | dat1;     // Los juntamos y devolvemos el valor
};
```

Una vez se dispone de esta función es sencillo resolver el ejercicio propuesto. Dado que hay que componer dos valores en el registro se puede pensar en componer un primer valor con ambos datos y llamar luego a la función o llamar dos veces a la función añadiendo cada vez uno de los valores. Veamos ambas posibles soluciones en este orden:

```
#define MASKPPAL 0xE0FC07FF // Mascara principal, 1 en los bits del registro
#define MASK7 0x7F // Mascara para dejar 7 bits a la derecha
#define MASK5 0x1F // Mascara para dejar 5 bits a la derecha

int registro, val1, val2;
//...
registro = lee_reg(REGXXX); // leemos el valor del registro de control
val1 = val1 & MASK7; // Dejamos solo 7 bits
val1 = val1 << 11; // y desplazamos al bit 11
val2 = val2 & MASK5; // Dejamos solo 5 bits
val2 = val2 << 24; // y desplazamos al bit 24
registro = mezcla(MASKPPAL, val1 | val2, registro); // Componemos
escribe_reg(REGXXX, registro); // Escribimos el nuevo valor en el
// registro de control
```

```
#define MASK7_11 0xFFFC07FF // Mascara para dejar 7 bits a partir del 11
#define MASK5_24 0xE0FFFFFF // Mascara para dejar 5 bits a partir del 24

int registro, val1, val2;
//...
registro = lee_reg(REGXXX); // leemos el valor del registro de control
registro = mezcla(MASK7_11, val1 << 11, registro); // Componemos el primer valor
registro = mezcla(MASK5_24, val2 << 24, registro); // Componemos el segundo valor
escribe_reg(REGXXX, registro); // Escribimos el nuevo valor en el
// registro de control
```