

Apellido: [REDACTED]
Titulación: 3º Graduado en Ingeniería Informática
Asignatura: SISTEMAS EMPOTRADOS
Fecha: 19 de enero de 2024

EXAMEN DE TEORÍA DE SISTEMAS EMPOTRADOS, 19 DE ENERO DE 2024.

- 1.- Describa los criterios que se deben seguir a la hora de seleccionar un determinado microcontrolador para el desarrollo de una aplicación. (1 punto).
- 2.- ¿Qué quiere decir que los procesadores ARM tienen arquitectura *Load-Store*? (1 punto).
- 3.- Explique los tres estados en que se puede encontrar el módulo acelerador de memoria. (1 punto). **MAM.**
- 4.- Conteste verdadero (V) o falso (F). Para que se considere verdadero toda la afirmación debe serlo. (1 punto).
 - ☐ Para desarrollar *software* en un sistema empotrado es imprescindible utilizar un sistema operativo para sistemas empotrados.
 - ☐ La diferencia entre la arquitectura *Harvard* y la arquitectura *Van Neuman* estriba en que en la *Harvard* puede haber memoria ROM y RAM, y en la *Van Neuman* sólo RAM.
 - ☒ En el procesador ARM7 se puede leer la palabra de estado independientemente del modo de privilegio.
 - ☐ Los procesadores ARM se consideran tipo RISC aunque tenga operaciones que procesen datos en memoria.
 - ☒ No todos los registros están disponibles cuando se utilizan instrucciones *Thumb* en los procesadores ARM7.
 - ☐ Todos los periféricos en la arquitectura ARM deben estar conectados al bus *Advanced Peripheral Bus* (APB).
 - ☒ A diferencia de los procesadores CISC, el ARM no soporta operaciones memoria a memoria.
 - ☒ El bus ASB (*Advanced System Bus*) puede sustituir al bus AHB (*Advanced High-performance Bus*) cuando no es preciso tan alto rendimiento.
 - ☒ El módulo acelerador de memoria (MAM) del LPC2378 es más complejo que una memoria caché.
 - ☐ El LPC2378 dispone sólo de dos fuentes de reloj: el cristal XTAL externo y el oscilador RC interno.
- 5.- Conteste verdadero (V) o falso (F). Considérese el procesador LPC2378. Para que se considere verdadero toda la afirmación debe serlo. (1 punto).
 - ☐ Tras un *reset* todos los periféricos tienen habilitado su reloj y se puede deshabilitar en el registro PCONP.
 - ☒ Soportar interrupciones multinivel (anidadas) en el LPC2378 sólo puede hacerse mediante la propia rutina de servicio a la interrupción (ISR).
 - ☒ Los registros especiales de función (SFR) disponen para su programación de tres registros: STATUS, CLEAR y SET.
 - ☐ La unidad de acceso directo a memoria (DMA) sólo puede acceder a datos de 32 bits.
 - ☐ Todos los registros de control de los GPIO están conectados al bus APB del LPC2378.
 - ☒ La fuente de reloj por defecto para todos los temporizadores es el reloj periférico PCLK del bus APB (*Advanced Peripheral Bus*).
 - ☒ El temporizador de vigilancia (WDT) puede seleccionar entre tres fuentes de reloj: oscilador interno, reloj en tiempo real (RTC) y el reloj de los periféricos PCLK.
 - ☐ La comparación del contador del *timer* del PWM se hace con el registro *match*, al igual que en los *timers* genéricos.



Apellidos y nombres:

Título: *Graduado en Ingeniería Informática, Ingeniería de Computación*

Asignatura: SISTEMAS EMPOTRADOS

Fecha: 19 de enero de 2024

ESCUELA POLITÉCNICA
SUPERIOR DE INGENIERÍA

- ✓ En el conversor analógico/digital (ADC) se tiene la opción de cambiar el número de bits por conversión.
- ✓ El conversor digital/análogo (DAC) dispone de varios registros para su programación.
- 6.- Conteste verdadero (V) o falso (F). Considere el procesador LPC2378. Para que considere verdadero toda la afirmación debe serlo. (1 punto).
- ✓ En una topología monomuestra, el muestreo ha de ser necesariamente el inicio de la transmisión automática.
- ✓ Las cuatro UARTs del microcontrolador LPC2378 disponen de un generador de reloj propio.
- ✓ Debido a la simpleza del bus FC, los dispositivos que conecte están obligados a disponer de propia señal de reloj.
- ✓ El bus SPI sólo contempla topologías con un solo maestro.
- ✓ El periférico SSP tiene una sola línea de interrupción conectada al VIK.
- ✓ El periférico I²S puede configurarse como un controlador de flujo DMA.
- ✓ El periférico SD-MMC tiene un bus de datos de ocho bits de ancho.
- ✓ Las tramas CAN2.0A y CAN2.0B no pueden coexistir en la misma red ya que tienen distinto formato.
- ✓ El controlador CAN del LPC2378 tiene tres buffers de transmisión y sólo uno de recepción.
- ✓ Un nodo Ethernet II no puede coexistir en la misma red con un nodo IEEE 802.3.
- 7.- Conteste verdadero (V) o falso (F). Para que se considere verdadero toda la afirmación debe serlo. (1 punto).
- ✓ El controlador Ethernet MAC del LPC2378 está ubicado en un bus AHB exclusivo para él.
- ✓ El bus USB es un puerto de comunicación full-duplex.
- ✓ El controlador USB del LPC2378 funciona con el reloj periférico PCLK.
- ✓ En el microcontrolador LPC2378 existen hasta cuatro fuentes distintas de reloj.
- ✓ Todas las transferencias de datos del controlador de memoria externa (EMAC) del LPC2378 deben tener 32 bits de ancho.
- ✓ El reloj en tiempo real (RTC) de que dispone el LPC2378 puede funcionar con el reloj de periféricos PCLK o con un oscilador externo.
- ✓ El planificador de sistema operativo *Instant Lip* solamente se puede aplicar a tareas periódicas.
- ✓ El método de utilizar "variables candado" en sistemas operativos no resuelve el problema de la exclusión mutua.
- ✓ El procesador Cortex-M3 soporta los modos de operación tradicionales del ARM y añade dos nuevos *Handler* y *Thread*.
- ✓ Los procesadores Cortex-M soportan las instrucciones Thumb de 16 bits al igual que el ARM.
- 8.- Describa la función de las tres señales del bus FS. (3 puntos)
- 9.- Indique los tipos de tramas CAN y cuándo se producen. (1 punto). Diez
- 10.- ¿Cuál es la principal diferencia entre un planificador cíclico y uno apropiativo? Describa las ventajas e inconvenientes de las estrategias de planificación cíclica. (3 puntos).

- 1.- En todas las prácticas hemos empleado un fichero `header`. Explique las definiciones que se incluyen en la parte de código siguiente y para qué se utilizan. ¿Cuáles son los registros del LPC2378 utilizados?. (1 punto)

```
#define PULSO0_LOW 5
#define PULSO0_HIGH 7
#define SIGNAL0_PIN_HIGH FIO4SET3 = 0x01;
#define SIGNAL0_PIN_LOW FIO4CLR3 = 0x01;
#include <LPC23xx.H>
```

- 2.- En la configuración de los temporizadores utilizamos la siguiente parte de código. Describa la utilidad de cada uno de los registros y qué se consigue en cada línea del código. (1 punto)

```
void delayT0Unlocked(unsigned int delayInDecimaMiliseg)
{
    TOTCR = 0x02;
    TOMR0 = delayInDecimaMiliseg * 12000000 / 10000;
    TOMCR = 0x07;
    TOTCR = 0x01;
}
```

- 3.- Describa la utilidad de la siguiente función y de cada uno de los registros que se utilizan en ella. (1 punto)

```
void pinesSignalInit(void)
{
    PINSEL9 = 0x00000000;
    PINMODE9 = 0x00000000;
    FIO4DIR3 = 0x03;
}
```

- 4.- En algunas prácticas utilizamos el controlador de interrupciones vectorizadas (VIC). ¿Qué ventajas tiene este hecho?. Describa la función siguiente y los registros utilizados. (1 punto)

```
void timer0Init(void)
{
    TOPR = 0x00;

    VICVectAddr4 = (unsigned long) T0_IRQHandler;
    VICVectCntl4 = 15;
    VICIntEnable = (1 << 4);
}
```

5.- En la configuración del puerto serie utilizamos el fichero `serial.c` de `Keil` del que la parte del código para configurar la UART se muestra a continuación. Responda a las siguientes cuestiones: (1 punto)

- Explique el hecho de que para definir la UART0 se utilice solo una línea de código para definir la UART1 se utilicen dos.
- ¿Por qué cambian los valores en `UxLCR` de `0x83` a `0x03`?
- ¿Qué significa el valor 78 en el registro `UxDLL`?

```
void init_serial(void) {
    #ifndef UART0
        PINSEL0 |= 0x00000050;
    #elif defined(UART1)
        PINSEL0 |= 0x40000000;
        PINSEL1 |= 0x00000001;
    #endif
    UxFDR = 0;
    UxLCR = 0x83;
    UxDLL = 78;
    UxDLM = 0;
    UxLCR = 0x03;
}
```

6.- En la práctica del conversor analógico/digital utilizamos la función siguiente. Indique el objetivo de cada una de las líneas del código. (1 punto)

```
void ADC_Init(void)
{
    PCONP      |= (1 << 12);
    PINSEL1     = 0x4000;
    AD0INTEN    = (1 << 0);
    AD0CR       = 0x00200301;
}
```

- ¿Qué se consigue y para qué se utiliza el fichero `hwinit_misTypes.h` que hemos empleado en todas las prácticas? ¿Qué hay que hacer para poder utilizarlo correctamente? (1 punto)
- ¿Cuál es la utilidad del fichero `HAL.c` utilizado en todas las prácticas? ¿Qué hemos incluido en él? (1 punto)
- Otro fichero que hemos utilizado en algunas prácticas ha sido el fichero `IRQ.c`. ¿Qué debe incluirse en él? ¿En qué prácticas se ha utilizado? (1 punto)
- Describa brevemente los objetivos de cada una de las prácticas realizadas. (1 punto)