



PROBLEMA 1 (2.5 puntos):

Diseñe un microprocesador con las siguientes características:

- 16KB de memoria de datos.
- 8KB de memoria de programa.
- Juego de 128 instrucciones de 16 bits.
- Bus de datos de 8 bits.
- Minimice la velocidad de carga de la instrucción (Fetch).

En caso de implementarse con arquitectura Von Neuman, se pide:

- a) Diagrama de bloques indicando el tamaño de los buses principales.
- b) Mapa de memoria.
- c) Tamaño del contador de programa y del registro de instrucción.

PROBLEMA 2 (2.5 puntos):

Se tiene un microprocesador con las siguientes características:

- Arquitectura Harvard
- 16 registros internos que pueden ser utilizados tanto para direcciones como para datos, sin poder acceder a la memoria de programa.
- 1KB de datos organizados en palabras de 8 bits
- Capacidad de direccionar 8K instrucciones
- Filosofía Load & Store

1. Diseñe una codificación de los distintos tipos de instrucciones microprocesador, minimizando en lo posible el tamaño de la palabra de instrucción. Los tipos de instrucciones que debe tener el microprocesador, son:

- 9 instrucciones aritmético/lógicas de operar entre registros, contemplando que el resultado siempre se guarda en uno de los dos registros de los operandos
 - Ejemplo: ADD Rd, Rs; $Rd \leftarrow Rd + Rs$
- 6 instrucciones aritmético/lógicas con direccionamiento inmediato, donde el resultado se almacenará en el registro del otro operando.
 - Ejemplo: ADDI Rd, 115; $Rd \leftarrow Rd + 115$
- 8 instrucciones de salto absoluto condicional
 - Ejemplo: JMP <direccion>; $PC \leftarrow <direccion>$
- 7 instrucciones de transferencia de información, con direccionamiento directo
 - Ejemplo: LD Rd, <direccion>; $Rd \leftarrow (<direccion>)$
- 7 instrucciones de transferencia de información, con direccionamiento indirecto
 - Ejemplo: LDI Rd, Ri; $Rd \leftarrow (Ri)$

2. Describa con el mayor detalle posible la ejecución de la siguiente instrucción en RTL, si se ejecutase en un microcontrolador con arquitectura de 32 bits y memoria de programa de 32 bits.

ADD Rd, Rs; $Rd \leftarrow Rd + Rs$



PROBLEMA 3 (5.0 puntos):

Muchos productos software utilizan “llaves hardware”, o lo que es lo mismo, dispositivos hardware que se conectan al ordenador para verificar que el producto software que está utilizando dispone de licencia. Actualmente trabaja usted en una empresa que le ha solicitado la implementación de un prototipo de “llave hardware” para puerto serie, es decir, un dispositivo que conectado al puerto serie de su ordenador verifique que el usuario dispone de una licencia.

La especificación indica que el PC enviará a la “llave hardware” un byte de control con 8 bits de datos, debiendo responder la “llave hardware” con el comando “ACK” en ASCII al recibir dicho comando.

Esto hará que el PC, en caso de no recibir “ACK” detecte que el programa está trabajando sin licencia (no obstante esto no tiene demasiada relevancia para usted dado que sólo le han encargado el diseño de la “llave hardware”).

El producto es autónomo, utilizando para su fuente de alimentación basada en pilas, que ofrece dos salidas a 3,3 V y 5 V.

Para la segunda reunión de proyecto, su superior le solicita:

- 1) Diagrama de bloques del sistema incluyendo todos los dispositivos (alimentación, adaptadores de nivel, osciladores, etc).
- 2) Diagrama de flujo de la aplicación (cuya recepción funcionará por interrupciones).
- 3) Rutina de configuración de la USART (incluir puertos e interrupciones): *void configUSART(void);*
- 4) Rutina de atención a la interrupción: *void USART1_IRQHandler(void);*
- 5) Rutina de envío de comandos por polling: *void sendTxt(char *dato);*
- 6) Programa principal: *int main(void);*