

1. Para los siguientes periféricos, describir brevemente su funcionamiento, los pasos a seguir para configurar e incluir tres aplicaciones reales de uso.

Periférico	Concepto	Configuración	Ejemplos
Puertos paralelos	Sirven como entrada o salida de información mediante señales analógicas o digitales	<ol style="list-style-type: none">1. Debe limpiarse el registro diseñado para almacenar el dato a enviar o recibir. Ejemplo: (PORTA, PORTB, etc)2. Utilizar el registro donde se configura el puerto Ejemplo: (TRISA, TRISB, etc) para configurar entradas o salidas. NOTA: Revisar que puertos pueden ser analógicos o digitales respecto al microprocesador3. Leer o escribir en el registro de datos del puerto.	<ul style="list-style-type: none">• Utilizar sensores para obtener información, infrarrojos, ultrasónicos• Controlar el encender o apagar leds
Convertidor A/D	Convierte una señal analógica a digital por medio de una resolución entre bits y rango de voltaje.	<ol style="list-style-type: none">1. Limpiar los puertos paralelos de entrada a usar2. Configurar los puertos analógicos dependiendo de qué puertos serán analógicos.3. Configurar la frecuencia4. Convertir5. Esperar conversión6. Obtener resultado y emplearlo para el uso que se quiera	<ul style="list-style-type: none">• Realizar voltímetros digitales• Convertir la señal analógica de distintos dispositivos como un potenciómetro a una señal digital
Puerto serie Asíncrono	Funciona con el objetivo de usar la comunicación serial asíncrona entre dos dispositivos del	<ol style="list-style-type: none">1. Seleccionar la velocidad (bandera BRGH)2. Cargar el valor de la velocidad requerida de acuerdo al microcontrolador.	<ul style="list-style-type: none">• Enviar datos mediante entre un teclado y dispositivos

	mismo tipo Enviando información con un baud rate por bit a bit	<ol style="list-style-type: none"> 3. Configurar modo asíncrono (bandera SYNC en TXSTA) 4. Habilitar transmisión (TXSTA) 5. Habilitar recepción 6. Habilitar el puerto serie 	<ul style="list-style-type: none"> • Conectar dispositivos inalámbricos
Comunicación I2C	<p>Es un tipo de comunicación que permite controlar dispositivos externos.</p> <p>Implementa maestros y esclavos para por medio de dos señales y un común.</p>	<ol style="list-style-type: none"> 1. Puede ser configurado de manera especial como esclavo. 2. Pueden configurarse dos puertos digitales para enviar las señales entre el dispositivo final y el PIC. Además de un bus de datos extra. 3. El bus de I2C posteriormente recibe o envía instrucciones y datos del bus de datos. 	<ul style="list-style-type: none"> • Conectar una pantalla lcd para controlarla • Emplear dispositivos externos con propósitos generales

2. .

- Explicar que es una interrupción
- Que se requisitos se deben cumplir para atender una interrupción
- El PIC16F877A, cuantas interrupciones tiene, listar 6 de ellas

a) Explicar que es una interrupción

Es una petición que se da al procesador para dejar de realizar la ejecución de un programa y atender otro evento que puede ser interno o externo.

b) Que se requisitos se deben cumplir para atender una interrupción

1. Habilitar la interrupción particular
2. Habilitar las interrupciones generales
3. Configurar el vector de interrupción
4. Incluir la rutina de interrupción

c) El PIC16F877A, cuantas interrupciones tiene, listar 6 de ellas

El pic tiene en total 15 interrupciones, algunos ejemplos son:

- Detección de flanco de RB0
- Captura y comparación de CCP1
- Escritura de byte en la memoria EEPROM

- Conversión completa en convertidor A/D
- Desbordamiento de TIMER1
- Desbordamiento de TIMER2

3. Representar el Pipeline en el microcontrolador PiC 16F877(A)

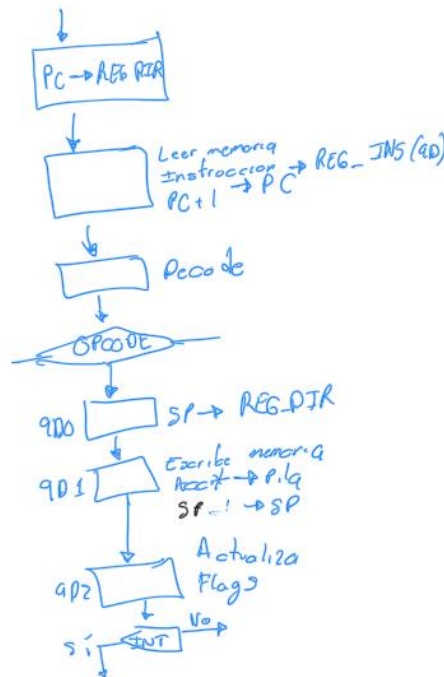
Debido a que el pic tiene una arquitectura Harvard, a continuación, se muestra un ejemplo de ejecución paralela de instrucciones en esta arquitectura

Ciclo	Búsqueda Instrucción	Decodificación	Búsqueda datos	Ejecución
4	I4	3	I2	I1
3	I3	I2	I1	
2	I2	I1		
1	I1			

4. Representar la carta ASM que describe las siguientes instrucciones, incluir en cada estado el contenido de los componentes involucrados

- PUSHA** ; guarda el contenido del acumulador A en la pila; OPCODE=0x9D (arquitectura Von Neuman)
- INCFZ F,d** ; incrementa el registro F y salta si es cero.

a) PushA



b) INCFZ F,d

(Faltó tiempo ☹)

5. Describir las diferencias que existen entre las arquitecturas Von Neuman y Harvard

1. La arquitectura Von Neuman ejecuta secuencialmente instrucciones mientras que la Harvard las hace de forma simultanea
2. Von Neuman tiene diversos modos de direccionamiento
3. Von Neuman son computadoras de conjuntos de instrucciones complejos (CISC) mientras que las Harvard son de de instrucciones reducidos (RISC)
4. Von Neuman tiene tamaños de instrucciones variables mientras que Harvard son fijas
5. Harvard posee la memoria de programa y de datos separada,

6. Se desea generar un contador a través de un puerto paralelo, con duración de 400 milisegundos entre cada estado; escribir las instrucciones que generan este tiempo:

a) Usando retardo mediante C

```
1  #include<16f877.h>
2  #fuses HS, NOPROTECT,
3  #use delay(clock=20000000)
4  #org 0x1F00, 0x1FFF void loader16F877 (void) {}
5
6  int main() {
7      int count = 0;
8      while (true) {
9          output_b(count);
10         delay_ms(400);
11         count++;
12     }
13     return 0;
14 }
```

b) Usando interrupciones del timer 2

```

1  #include<16f877.h>
2
3  #fuses HS, NOWDT, NOPROTECT, NOLVP
4  #use delay(clock=2000000)
5
6  #Global variables
7  int count = 0;
8  int time= 0;
9
10 #int_rtcc
11 clock_isr() {
12     if(time == 19) {
13         output_b(count);
14         count++;
15         time = 0;
16     }
17     time++;
18 }
19
20 void main() {
21     set_timer2(0);
22     setup_counters(RTCC_INTERNAL, RTCC_DIV_256);
23     enable_interrupts(INT_RTCC);
24     enable_interrupts(GLOBAL);
25     while(1) {
26     }
27 }

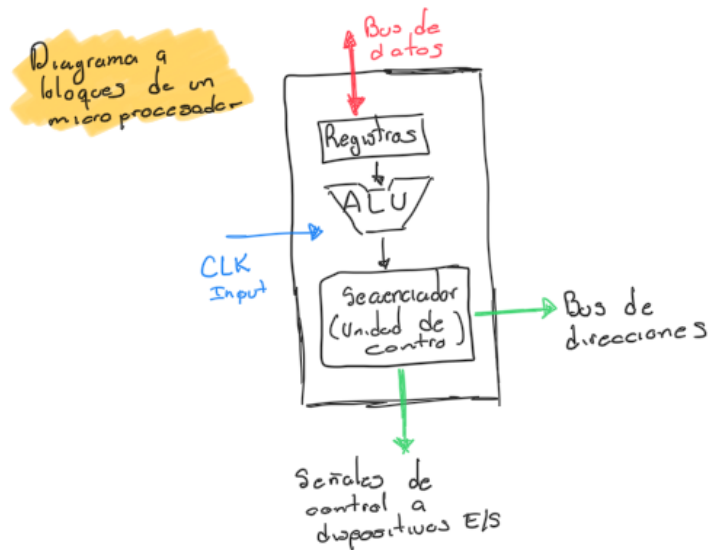
```

7. Definir que es un microprocesador, agregar diagrama y componentes internos.

Es una unidad de propósito general que se procesa y controla las tareas, cuenta para ello con distintos bloques (Ver imagen inferior) que llevan a cabo el ciclo fetch.

Componente:

- **Registros internos:** Es donde podemos encontrar el código de instrucciones. Elemento con información con tamaño definido.
- **ALU (Unidad Lógica Aritmética):** que realiza las operaciones lógicas y aritméticas. De aquí se obtiene los datos del bus de datos.
- **Secuenciador o Unidad de Control:** Es la encargada de generar y administrar el orden de las acciones.



8. Que es el ciclo fetch y de que se compone.

Ciclo Fetch o también ciclo de instrucción, es todo el período que tarda la unidad de procesamiento en procesar una instrucción, además se compone de 4 etapas:

1. Búsqueda de la instrucción:
 - se realiza la búsqueda de la instrucción en memoria y se envía al registro de instrucciones.
2. Decodificación:
 - Se obtiene el código a realizar de la instrucción (OPCODE)
3. Búsqueda de datos:
 - Obtenemos los operandos de los registros necesarios en memoria.
4. Ejecución de la instrucción:
 - De acuerdo al opcode y los operandos encontrados se realiza la definición de la operación.

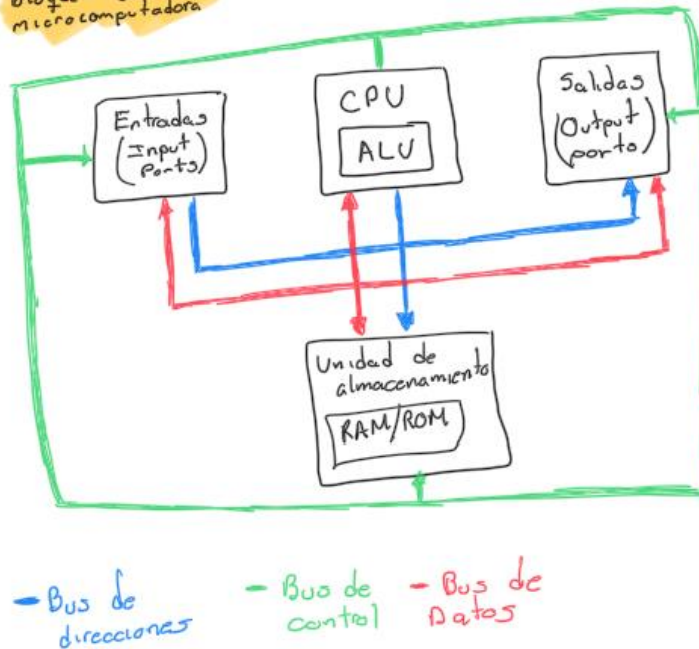
9. Definir que es una microcomputadora y cual es la diferencia con respecto a un microcontrolador, agregar diagramas.

Una microcomputadora es una entidad de propósito general que contiene diversas unidades destinadas a la manipulación y coordinación de los procesos que esta debe llevar a cabo, es el caso de puertos de entrada y salida, unidades de procesamiento o incluso unidades de almacenamiento

Algunas diferencias entre una microcomputadora y un microcontrolador:

1. El propósito al que están destinados, por ejemplo, no es lo mismo un Arduino o un ESP respecto a un Raspberry. Los microcontroladores son sistemas embebidos de bajo precio que usualmente son más para tareas específicas en cambio una microcomputadora es de propósito general
2. Los microcontroladores no tienen sistemas operativos mientras que las microcomputadoras si lo tienen, es decir, la gestión de procesos, y la forma en que se coordina todo en las microcomputadoras se tiene una capa de abstracción sujeta a un software
3. La cantidad de memoria y el poder de procesamiento es mayor en una microcomputadora en general.

Diagrama a bloques de una microcomputadora



10. En un programa se tiene la siguiente asignación: `DATO EQU 0X20`; indicar que se obtiene con las siguientes instrucciones (todas deben estar correcta para obtener un punto).

- `MOVWF DATO`
- `MOVF DATO.W`
- `MOVLW DATO`
- `GOTO DATO`
- `GOTO $-DATO`

- Se mueve el contenido del registro W a DATO (El registro 0x20)
- Se mueve el contenido del registro DATO al registro W
- Se mueve la constante 0x20 al registro W
- Se salta a la etiqueta DATA y se carga en PC la constante 0x20, además de que se ejecuta la instrucción
- Se regresa 0x20 instrucciones desde la instrucción presente o actual