

Prototipo de microcontrolador de aplicación específica

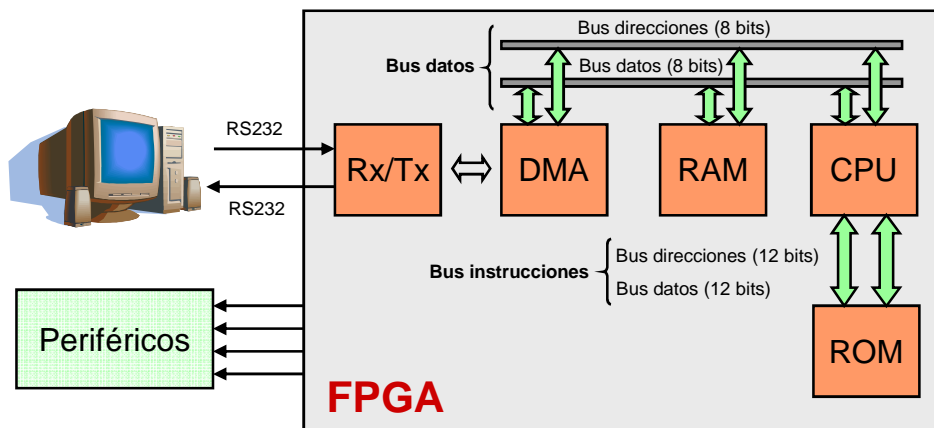
Práctica III

DSED Curso 2014-2015

Introducción: objetivos

✓ Objetivo → **Desarrollo de un microcontrolador**

- Escrito en VHDL
- Implementado sobre FPGA



Introducción: componentes

- ✓ Periféricos gestionados
 - 8 interruptores ON/OFF
 - 10 actuadores de nivel
 - Un termostato
- ✓ Interfaz serie (RS232) de comunicación
 - μ controlador recibe órdenes a través de línea RS232
 - μ controlador envía información por línea RS232
- ✓ Recepción y envío de datos
 - Controlado por DMA
 - Toma control de los buses
 - Almacena/lee datos en/de RAM
 - Necesario protocolo comunicación DMA- μ controlador

3

Comandos del terminal (I)

Comando	Parámetro 1	Parámetro 2	Descripción
I	0..7	0, 1	Selecciona un interruptor de los 8 existentes y lo enciende (1) o lo apaga (0)
A	0..9	0..9	Selecciona uno de los 10 actuadores y le asigna un valor de apertura de 0 a 9
T	1, 2	0..9	Carga el valor del termostato a una temperatura entre 10 y 29 grados
S	I, A, T	0..9	Solicita información al procesador sobre el estado de alguno de los periféricos

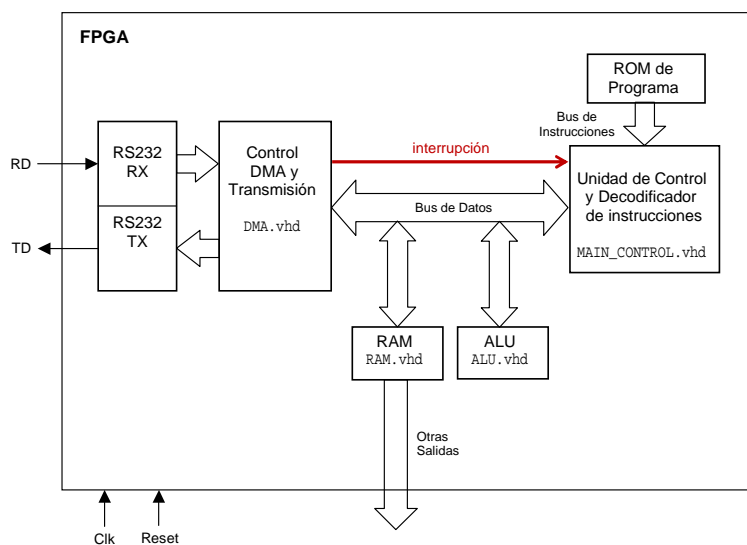
4

Comandos del terminal (y II)

- ✓ Decodificación de comandos del terminal:
 - Los comandos del terminal se procesan mediante **programa** en el μ controlador
 - Se proporcionará:
 - Programa para interpretar los comandos del terminal
→ almacenado en ROM
 - Compilador-ensamblador (**uso opcional**)

5

Arquitectura e interfaz externa



6

Interfaz RS-232 (I)

- ✓ RX y TX a 115.200 bps.
- ✓ Formato trama: 8 bits datos, sin paridad, bit parada
- ✓ FIFO 16 bytes para almacenamiento datos recibidos
- ✓ Indicadores de memoria llena y memoria vacía
- ✓ Entrega de datos recibidos (*recepción*):
 - Interfaz paralela y síncrona (línea de Data_Read)
- ✓ Captura de dato (*transmisión*):
 - Interfaz asíncrona
 - Línea de validación de datos (Valid_D) y de reconocimiento de carga (ACK_in)
 - Línea de estado del transmisor (TX_RDY)

7

Interfaz RS-232 (y II)

LÍNEA	SENTIDO	FUNCIÓN
Reset	Entrada	Reset asíncrono y activo a nivel bajo.
Clk	Entrada	Reloj del sistema.
Data_in [7..0]	Entrada	Entrada de los datos que el sistema cliente desea enviar. Bit más significativo en la posición 7.
Valid_D	Entrada	Validación del dato de entrada por parte del sistema cliente. Activa a nivel bajo.
ACK_in	Salida	Reconocimiento del dato para transmitir. Activa a nivel bajo, se activa al guardar un nuevo dato válido y se desactiva al desactivarse Valid_D.
TX_RDY	Salida	Transmisor está disponible. Activa a nivel alto.
TD	Salida	Línea de transmisión RS232.
RD	Entrada	Línea de recepción RS232.
Data_out [7..0]	Salida	Datos de salida de la FIFO interna del subsistema receptor. Bit más significativo en la posición 7.
Data_read	Entrada	Petición del sistema cliente de lectura de un nuevo dato de los recibidos y almacenados. Activa a nivel alto y síncrona con el reloj.
Full	Salida	Indica que la memoria interna del subsistema receptor está llena. Activa a nivel alto.
Empty	Salida	Indica que la memoria interna del subsistema receptor está vacía. Activa a nivel alto.

8

Mapa de memoria (I)

Dirección	Alias	Función
0x00	DMA_RX_Buffer (MSB)	Byte más significativo de la reserva para el controlador DMA (recepción)
0x01	DMA_RX_Buffer	Byte intermedio de la reserva para el controlador DMA (recepción)
0x02	DMA_RX_Buffer (LSB)	Byte menos significativo de la reserva para el controlador DMA (recepción)
0x03	NEW_INST	Flag que indica la llegada de un nuevo comando por la línea serie
0x04	DMA_TX_Buffer (MSB)	Byte más significativo de la reserva para el controlador DMA (transmisión)
0x05	DMA_TX_Buffer (LSB)	Byte menos significativo de la reserva para el controlador DMA (transmisión)
0x06 ... 0x0F	Reservado	Para posterior ampliación
0x10 ... 0x17	SWITCH(0..7)	Zona de control de interruptores
0x18 ... 0x1F	Reservado	Para posterior ampliación
0x20 ... 0x29	LEVER(0..9)	Zona de control de actuadores
0x2A ... 0x30	Reservado	Para posterior ampliación
0x31	T_STAT	Temperatura fijada en el termostato
0x32 ... 0x3F	Reservado	Para posterior ampliación
0x40 ... 0xFF	GP_RAM	Memoria de propósito general

9

Mapa de memoria (y II): interfaz

Línea	Sentido	Descripción
Reset	Entrada	Reset asíncrono a nivel bajo para los registros de uso específico de la RAM
Clk	Entrada	Reloj principal del sistema (20MHz)
Databus[7..0]	Bidireccional	Bus de datos del sistema
Address[7..0]	Entrada	Direcciones del bus de datos
CS	Entrada	Chip Select no considerar este puerto
Write_en	Entrada	Habilitación de escritura (activa junto con el flanco de subida del reloj)
OE	Entrada	Habilitación de lectura
Switches[7..0]	Salida	Estado de los interruptores
Temp_L[6..0]	Salida	dígito más bajo del valor de la temperatura del termostato (formato 7 segmentos)
Temp_H[6..0]	Salida	dígito más bajo del valor de la temperatura del termostato (formato 7 segmentos)

10

Controlador de DMA (I): funcionalidad

- ✓ Volcar a memoria los datos llegados a la interfaz RS232
 - Pedir buses al procesador
 - Cargar los bytes desde la FIFO del RS232 en las direcciones de memoria habilitadas al efecto (3 bytes: 0x00 a 0x02)
 - **Activar la línea de interrupción**
- ✓ Cargar en la interfaz RS232 los datos a transmitir
 - Comandos de 2 bytes (0x04 y 0x05)
 - Transmisión iniciada por una instrucción especial del procesador

11

Controlador de DMA (II): interfaz

Línea	Sentido	Descripción
Reset	Entrada	Reset asíncrono y activo a nivel bajo
Clk	Entrada	Reloj principal del sistema (20MHz)
RCVD_Data[7..0]	Entrada	Dato recibido por la línea 232
RX_Full	Entrada	Señal de estatus de la memoria interna del receptor
RX_Empty	Entrada	Señal de estatus de la memoria interna del receptor
Data_Read	Salida	Petición de lectura de un nuevo dato de los recibidos
ACK_out	Entrada	Señal de reconocimiento de llegada de datos al Transmisor RS232
TX_RDY	Entrada	Estatus de la máquina de transmisión serie
Valid_D	Salida	Validación del dato enviado al transmisor RS232
TX_Data [7..0]	Salida	Dato para enviar por línea serie
Address [7..0]	Salida	Direcciones del bus de datos del sistema
Databus [7..0]	Bidireccional	Bus de datos del sistema
CS	Salida	Chip select de la no considerar este puerto
Write_en	Salida	Habilitación de escritura para la RAM
OE	Salida	Habilitación de la salida de la RAM
DMA_RQ	Salida	Petición de buses al procesador principal
DMA_ACK	Entrada	Reconocimiento y préstamo de buses por parte Del procesador principal
Send_comm	Entrada	Señal de comienzo de envío de datos, controlada Por el procesador principal
READY	Salida	Señal a nivel alto únicamente cuando el procesador se encuentre totalmente ocioso (Idle)
int_RQ	Salida	Petición de interrupción

12

DMA (y III): secuencia de operaciones

✓ Recepción

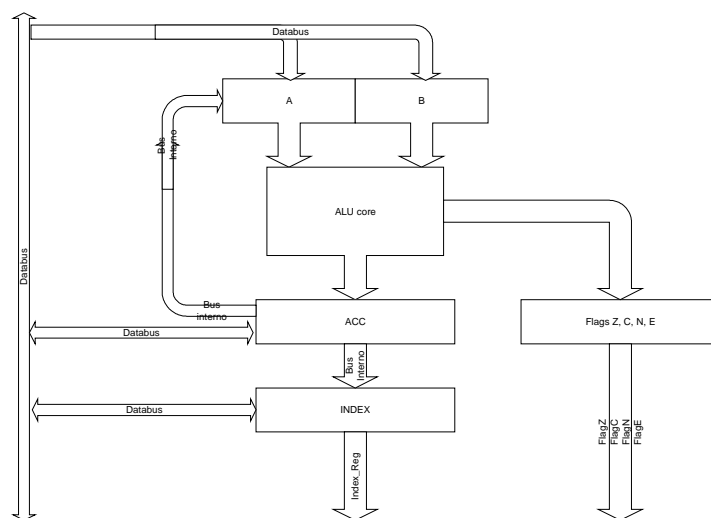
1. Controlador coloca un '1' en la línea DMA_RQ
2. El procesador concede los buses y coloca un '1' en la línea DMA_ACK
3. Controlador realizar la escritura en la RAM de acuerdo con la temporización que ésta ofrece
4. Controlador coloca un '0' en la línea DMA_RQ para devolver el control de los buses al procesador
5. El procesador coloca un '0' en la línea DMA_ACK y continúa la ejecución

✓ Transmisión

1. El procesador pone a '1' la línea SEND_comm, lo que fuerza que la línea READY se coloque a nivel bajo
2. Controlador realiza la transmisión de los 2 bytes, uno tras otro (está en posesión de los buses)
3. Terminada la transmisión de los datos, el controlador coloca a nivel alto la línea READY
4. El procesador puede retirar la petición de transmisión

13

Unidad Aritmético-Lógica (I): arquitectura



14

Unidad Aritmético-Lógica (y II): interfaz

Línea	Sentido	Descripción
Reset	Entrada	Asíncrono y activo a nivel bajo. Inicializa todos los registros internos a 0x00
Clk	Entrada	Reloj principal del sistema, 20MHz
Alu_op[5..0]	Entrada	Bus de microinstrucciones del procesador. La ALU permite hasta 64 operaciones, en el este prototipo sólo se implementarán algunas
Databus	Bidireccional	Bus de datos del sistema
Index_Reg	Salida	Conexión directa del registro de índice a la unidad decodificadora
FlagZ	Salida	Flag de cero
FlagC	Salida	Flag de acarreo
FlagN	Salida	Flag de acarreo en 'nibble'
FlagE	Salida	Flag de error

no implementar estos flags

15

ROM de programa

- ✓ Almacena el programa que ejecutará el μ controlador
- ✓ Las instrucciones son de longitud variable:
 - 1 ó 2 palabras de 12 bits
 - Primera palabra define la **funcionalidad**
 - Segunda palabra define **constantes o direcciones de memoria** en instrucciones que lo requieran
 - Por ejemplo en una instrucción de salto

Línea	Sentido	Descripción
INS_Addr[11..0]	Entrada	Bus de direcciones de la ROM de programa
INS_Bus[11..0]	Salida	Bus de instrucciones

16

Unidad de control (I): secuencia de operaciones

1. Si el DMA pide buses:
 - Concedérselos inmediatamente
 - Detenerse hasta que el DMA acabe su acceso a memoria
2. **Atender interrupciones**
3. Recoger una palabra de la memoria (**Fetch**)
4. Decodificar la instrucción correspondiente (**Decode**)
 - PIC_pkg.vhd define constantes para instrucciones
5. Ejecutar la instrucción (**Execute**):
 - Recoger una nueva palabra (si es necesario)
 - Generar las microinstrucciones adecuadas
6. Regresar al estado inicial

17

Unidad de control (II): componentes

- ✓ Autómata de control
 - Fetch, decode, execute...
- ✓ Componentes auxiliares
 - Contador de programa:
 - Dirección en ROM de la próxima instrucción
 - Registro de instrucciones:
 - Copia interna de la instrucción que se está ejecutando
 - **Registros para atención de interrupciones:**
 - **Registro para salvar la dirección de retorno**

18

Unidad de control (y III): interfaz

Línea	Sentido	Descripción
Reset	Entrada	Asíncrono y activo a nivel bajo
Clk	Entrada	Reloj principal del sistema, 20MHz
ROM_Data[11..0]	Entrada	Bus de datos de la memoria del programa
ROM_Addr[11..0]	Salida	Bus de direcciones de la memoria de programa
RAM_Addr	Salida	Bus de direcciones de la memoria de datos
RAM_CS	Salida	Chip select de la RAM no considerar este puerto
RAM_Write	Salida	Microinstrucción para escritura en la RAM
RAM_OE	Salida	Microinstrucción para permitir la salida de datos de la RAM
Databus	Bidireccional	Bus de datos del sistema
DMA_RQ	Entrada	Línea de petición de buses del controlador DMA
DMA_ACK	Salida	Microinstrucción de entrega de los buses al controlador DMA
SEND_comm	Salida	Microinstrucción para iniciar una transmisión por la línea serie
DMA_READY	Entrada	Señal de estado del controlador DMA
ALU_op[5..0]	Salida	Microinstrucciones con la operación parar realizar en la ALU
FlagZ	Entrada	Flag de cero de la ALU

int_RQ	Entrada	Petición de interrupción
--------	---------	--------------------------

19

Atención de interrupciones

1. Almacenar dirección de retorno y registros necesarios
2. Escribir el valor 0xFF en el registro NEW_INST de la RAM
3. Recuperar registros
4. Borrar la interrupción
5. Volver a la dirección almacenada

20

Instrucciones del μ controlador (I)

✓ Tipo 1, instrucciones para la ALU

- 1 byte
- "0000" & TYPE_1(2) & OP(6)

Nemónico	Descripción	Flags que modifica
ADD	A + B	Z, C, N
SUB	A - B	Z, C, N
SHIFTL	Gira hacia la izquierda el contenido del acumulador, introduciendo un cero	
SHIFTR	Gira hacia la izquierda el contenido del acumulador, introduciendo un cero	
AND	'and' lógico entre A y B	Z
OR	'or' lógico entre A y B	Z
XOR	'xor' lógico entre A y B	Z
CMPE	A = B	Z
CMPG	A > B	Z
CMPL	A < B	Z
ASCII2BIN	Convierte A del formato ASCII al binario (para números, devuelve FF si hay error)	E
BIN2ASCII	Convierte A del binario al ASCII (para números menores de 0x10, devuelve FF si hay error)	E

21

Instrucciones del μ controlador (II)

✓ Tipo 2, instrucciones de salto

- 2 bytes
- "0000" & TYPE_2(2) & OP(6)
- 2º byte: destino (8 bits)

Nemónico	Parámetro	Descripción
JMP	Dirección Inmediata Etiqueta (#Nombre)	Salto incondicional
JMPTrue	Dirección Inmediata Etiqueta (#Nombre)	Salto si FlagZ = '1'
RETI	Se ignora	Retorno de interrupción

22

Instrucciones del μ controlador (III)

- ✓ Tipo 3, movimiento de datos
 - 1 byte si la transferencia es entre registros
 - 2 bytes en el resto de casos

Nemónico	Parámetro Destino	Parámetro Origen	Descripción
LD LDI	.A .B .INDEX	.ACC	Movimiento entre registros
LD	.A .B .INDEX .ACC	Constante [Constante] [Dirección Inmediata]	Carga los registros de la ALU
LDI	.A .B .INDEX .ACC	[Constante] [Dirección Inmediata]	Carga los registros en la ALU, utiliza el registro .INDEX como índice para acceder a memoria
WR	Dirección inmediata Constante		Escribe datos en memoria
WRI	Dirección inmediata Constante		Escribe datos en memoria, utiliza el registro .INDEX como índice para acceder a memoria

23

Instrucciones del μ controlador (y IV)

- ✓ Tipo 4, instrucción para transmisión
 - 1 byte
 - "0000" & TYPE_4(2) & "000000"

24

Planificación

Se proporcionará el código del control principal

	Sistemas desarrollados
Etapas 1	RAM y decodificación de direcciones, pruebas
Etapas 2	DMA integrado con RS232 y RAM, pruebas
Etapas 3	ALU, pruebas
Etapas 4	Integración con la CPU, pruebas
Etapas 5	Simulación post-route
Etapas 6	Atención de interrupciones, pruebas