

# CC4301 Arquitectura de Computadores

## Solución Control 3

Profesor: Pablo Guerrero P.  
Prof. Aux.: Gaspar Pizarro V.  
Ayudante: Ian Yon Y.

30 de noviembre de 2012

### 1. Conceptos

- I Un controlador es un microprocesador dedicado a una función específica requerida para el funcionamiento de algún dispositivo, que intercambia información con la CPU a través de una línea de comunicación. Estos dispositivos se necesitan porque en un dispositivo E/S el programa de control acapara todo el tiempo de cómputo de la CPU, de forma que con un controlador E/S el trabajo de controlar un dispositivo se reduce a comunicarse con el controlador, liberando la CPU.
- II 1) Un dispositivo E/S pone en 1 su línea de interrupción, conectada a una de las líneas de interrupción de la CPU.  
2) La CPU suspende la ejecución del proceso en curso e invoca la rutina de atención de interrupciones.  
3) La rutina trabaja, interactuando con el dispositivo.  
4) La CPU retorna el control al proceso interrumpido, sin que este sepa que este fue interrumpido.
- III El registro AR en M32 es un registro asíncrono que mantiene el bus de direcciones con direcciones “útiles”, esto es que cuando se pide una dirección en memoria, el latch AR permite mantener la dirección mientras la memoria responde. El registro IR es un registro síncrono cuya función es mantener la instrucción actual ejecutándose, para que el resto de la CPU pueda disponer de su información (registros involucrados, valores inmediatos, opcodes, etc).
- IV Facilidad en la transmisión: Un cable paralelo no puede ser demasiado largo, y es más caro que un cable serial. Por otra parte la recepción de datos de forma paralela se complica porque no todos los bits llegan al mismo tiempo. Esto incrementa los costos de la transmisión paralela frente a la serial.
- V La ALU realiza las operaciones aritméticas y lógicas del computador. La unidad de control es un circuito secuencial que fija las señales de control en cada ciclo para los componentes tanto internos como externos, de forma de ejecutar las instrucciones correctamente.
- VI
  - a) RD. Es fijado por la CPU, e indica a la memoria que se va a leer un dato.
  - b) WR. Es fijado por la CPU, e indica a la memoria que se va a escribir un dato.
  - c) WAIT. Es fijado por la memoria, e indica que el dato en el bus de datos aún no está listo.

### Asignación de Puntaje

Definida en enunciado.

## 2. CPU

a

Ciclo	Transferencias entre registros	Seales de Control
Fetch1	AR:=PC goto Fetch2	OP-Y-SEL:=@0 OP-ALU:=@OR WR-AR; EN-A; OP-ABI:=@W
Fetch2	IR:= $Mem^w[AR]$ if WAIT goto Fetch2 else goto decode	OP-DBI:=@LDW SEL-D, WR-IR, EN-A, RD OP-ABI:=@W
Decode	PC:=PC+4 goto Exec1	OP-Y-SEL:=@4 OP-ALU:=@ADD WR-PC

Ciclo	Transferencias entre registros	Señales de Control
Exec1: ADD R1, 100, R2	R2:= R1+100 goto Fetch1	SEL-REG, WR-RD, WR-SR, RD-DEST OP-ALU:=ADD OP-Y-SEL:=@INST
Exec1: LDSB [R5+20], R3	AR:=R5+20 goto Exec2	SEL-REG, WR-AR, EN-A OP-ABI:=@B OP-ALU:=@ADD OP-Y-SEL:=@INST
Exec2: LDSB [R5+20], R3	R3:= $Ext_s(Mem^b[R5 + 20])$ if WAIT goto Exec2 else goto Fetch1	OP-ABI:=@B OP-DBI:=@LDSB SEL-D, WR-RD, EN-A

b

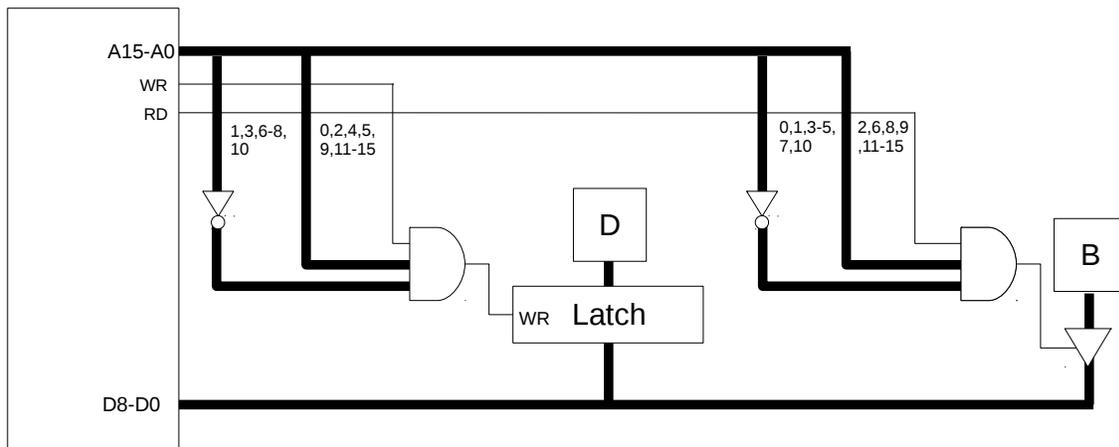
Inst	Opcode	$C_{in}$	C	SEL	Y/ Y
ADD	0000	X	0	110	1
ADDX	0001	0	0	110	1
	0001	1	1	110	1
SUB	0010	0	1	110	0
	0010	1	1	110	0
SUBX	0011	0	0	110	0
	0011	1	1	110	0
AND	0100	X	1	000	1
OR	0101	X	0	001	1

### 2.1. Asignación de Puntaje

- a)
- 0,1 por ciclos Fetch1, Fetch2, Decode.
  - 0,3 por instrucción ADD.
  - 0,4 por instrucción LDSB.
- b) 0,25 por cada columna distinta de  $C_{in}$ .

### 3. Entrada/Salida

a



```
void p3b()
{
    char *port_display = (char *) 0xFA35;
    char *port_button = (char *) 0xFB44;
    char *port_data = (char *) 0xF12C;
    char *port_ctrl = (char *) 0xF12D;
    char est;
    while (!(*port_ctrl&1));
    *port_display = getBytePattern(*port_data);
    est = *port_button;
    if (est) {
        while (!(*port_ctrl&2));
        *port_data = 0x00FF;
    }
}
```

```
void p3c()
{
    while (!0) p3b();
}
```

```
void p3d()
{
    char *port_data = (char *) 0xF12C;
    char *port_ctrl = (char *) 0xF12D;
    int x = 0;
    while (!0) {
        while (!(*port_ctrl&1));
        if (*port_data==0x00FF) {
            x++;
            while (!(*port_ctrl&2));
            *port_data = x;
        }
    }
}
```

```
}  
  
void p3e()  
{  
    char *port_data = (char*)0xF12C;  
    char *port_ctrl = (char*)0xF12D;  
    static int x;  
    x++;  
    while (!(*port_ctrl&2));  
    *port_data = x;  
}
```

### Asignación de Puntaje

- a) 0,5
- b) 0,5
- c) 0,2
- d) 0,4
- e) 0,4

La ausencia de busy-wait para poder leer/escribir en puerto serial tiene descuento de 0,1 puntos.