

Pauta Auxiliar 8

Profesor: Pablo Guerrero.

Auxiliar: Ian Yon

Martes 4 de agosto de 2015



Departamento de Ciencias de la Computación
UNIVERSIDAD DE CHILE

Problema 1

1. No es posible implementar SWAP en M32 como una sola instrucción porque falta un registro para almacenar el valor temporal de intercambio.
2. Crearemos un componente XR que será un registro auxiliar que funcionará igual que PC o AR (lo haremos funcionar como un LATCH) y agregaremos un multiplexor a M32. Además agregaremos las siguientes señales de control: *wr-xr*, que controlará si se escribe en XR o no; y *sel-reg-aux*, que servirá para que el multiplexor sepa cuando usar la información de XR.

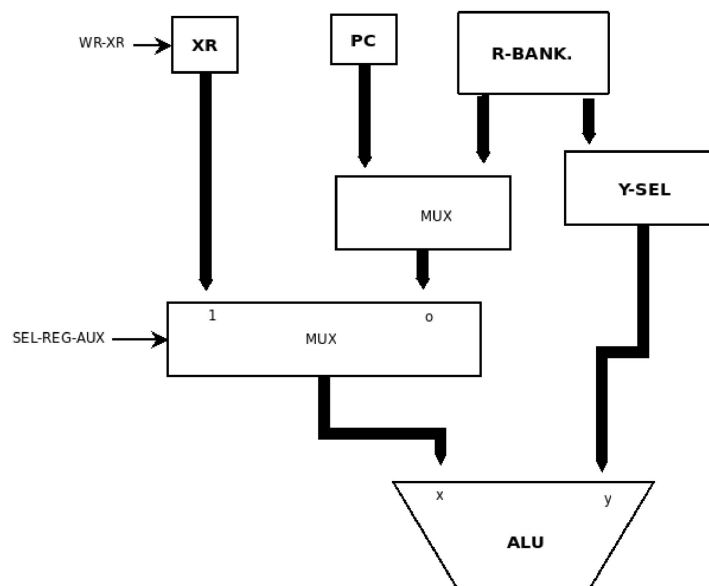


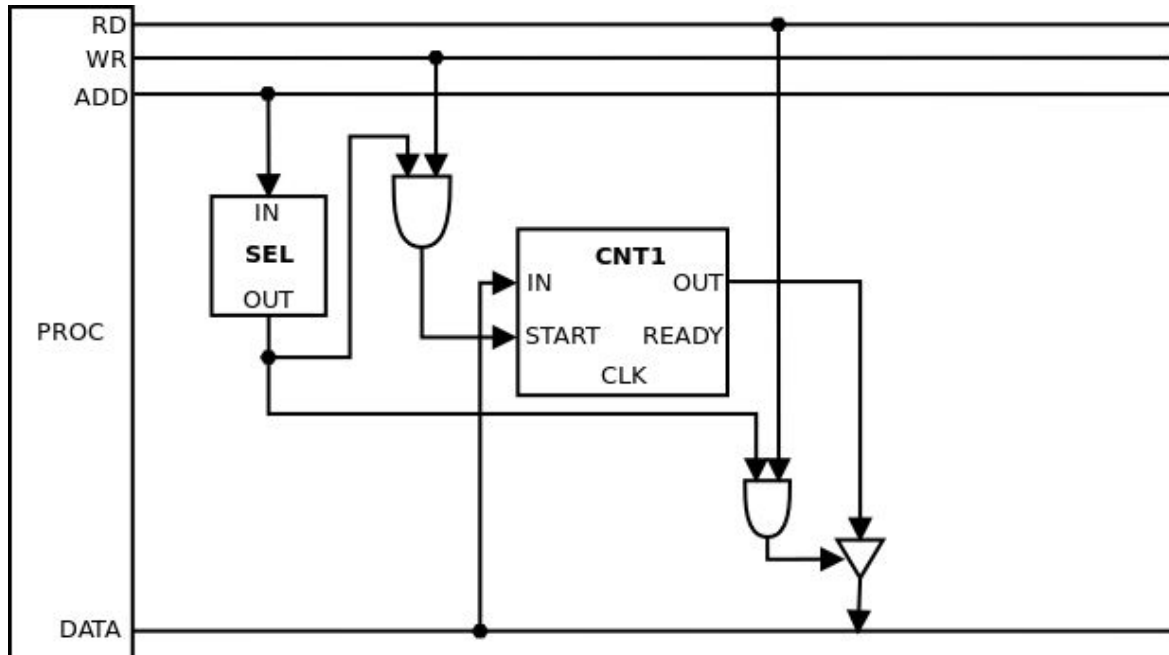
Figura 1: Modificaciones propuestas

3. Las señales de control

Ciclo	Acción	Señales
Execute 1	$AR \leftarrow reg_1 + \{imm reg_2\}$ $A31-A2 \leftarrow AR$	$SEL-REG, OP-Y-SEL \leftarrow @Inst, OP-ALU \leftarrow @ADD, WR-AR, OP-ABI \leftarrow @w, EN-A$
Execute 2	$XR \leftarrow M^w[AR]$	$RD, WR-XR, SEL-REG-AUX, SEL-D, OP-DBI \leftarrow @LDW, OP-ABI \leftarrow @W, EN-A$
Execute 3	$M^w[AR] \leftarrow reg_d$	$RD-DEST, OP-Y-SEL \leftarrow @0, OP-ALU \leftarrow @OR, OP-DBI \leftarrow @STW, OP-ABI \leftarrow @W, EN-A, EN-D, WR$
Execute 4	$reg_d \leftarrow XR$	$OP-Y-SEL \leftarrow @0, WR-RD, OP-ALU \leftarrow @OR, SEL-REG-AUX$

Problema 2

1. El diagrama de conexión queda como sigue:



Observación: El circuito SEL pone su salida en 1 si su entrada es 0x30F1A000. En caso contrario, su salida es 0.

2. La rutina para utilizar este circuito será:

```
int contar(int val){
    int i;
    int *port = 0x30F1A000;
    *port = val;
    for(i=0; i<8; i++);
    return *port;
}
```

Problema 3

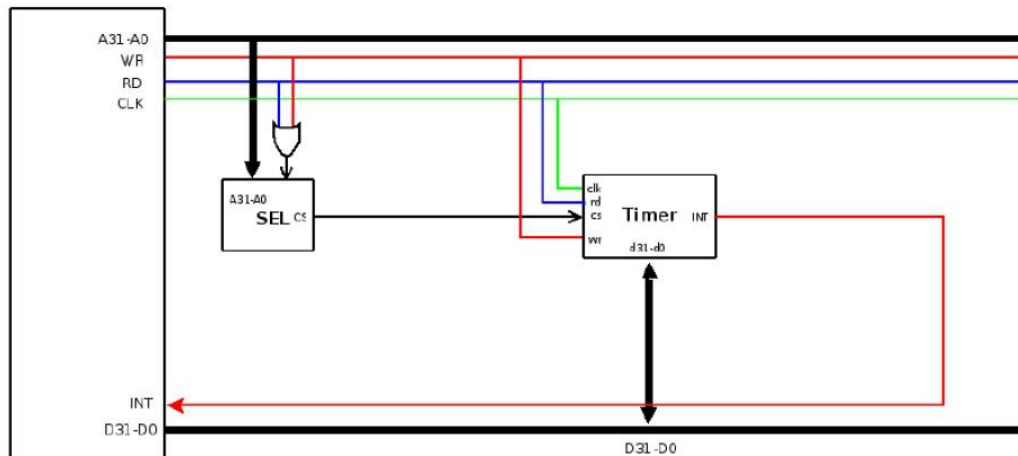


Figura 3: Interfaz TIMER

La salida del circuito SEL, CS, vale 1 si la dirección vale 0xffff0000 y RD o WR vale 1, es decir, si leo o escribo en la dirección 0xffff0000.

Procedimientos:

```
void *f_handler;

progTimer(int t, void (* f) ()){
    char *port = (char *)0xffff0000;
    *port = t;
    *int_handler = f;
}

handleTimer(){
    char *port = (char *) 0xffff0000;
    *port = 0;
    (*int_handler)();
}
```