BUSES Y ASM

Técnicas Digitales I

Luis Eduardo Toledo

UTN

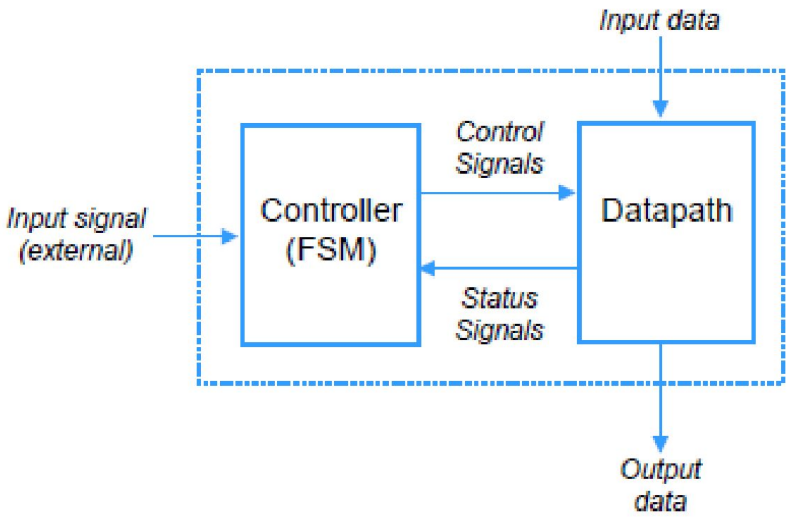
DESCOMPOSICIÓN EN UNIDAD DE PROCESO Y UNIDAD DE CONTROL

La información binaria en los sistemas digitales se puede clasificar en dos categorías:

DATO

Elementos discretos de información manipulados por operaciones aritmética, lógica, de desplazamiento y otro procesamiento de datos.

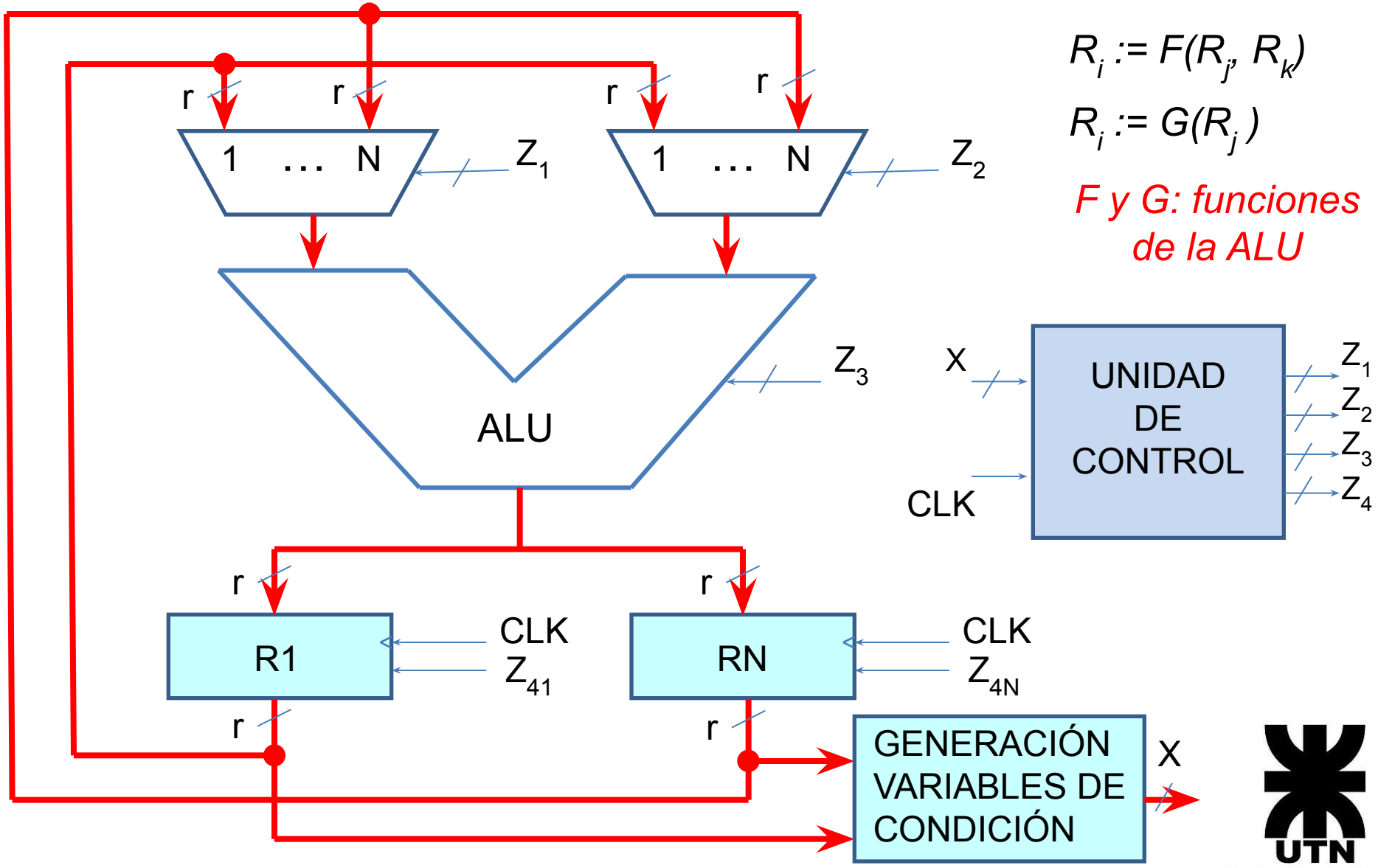
Operaciones implementadas a través de componentes digitales como sumadores, decodificadores, multiplexores, etc.



CONTROL

Proporciona señales de comando que coordinan la ejecución de varias operaciones en la sección de datos para lograr la tarea deseada.

MÁQUINA ALGORÍTMICA ESTÁNDAR



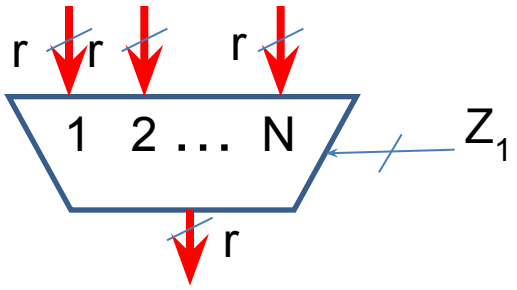
$R_i := F(R_j, R_k)$

$R_i := G(R_j)$

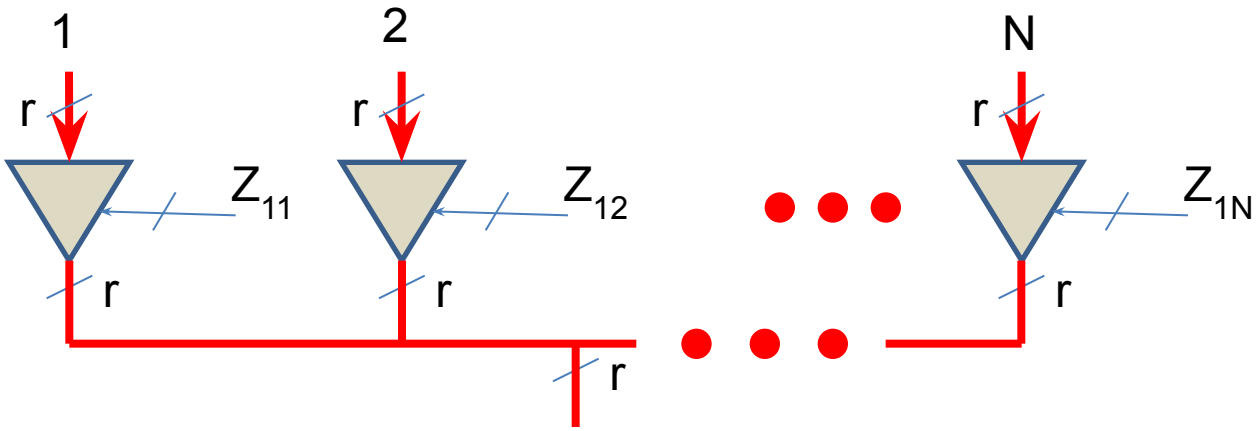
F y G: funciones de la ALU

UTN

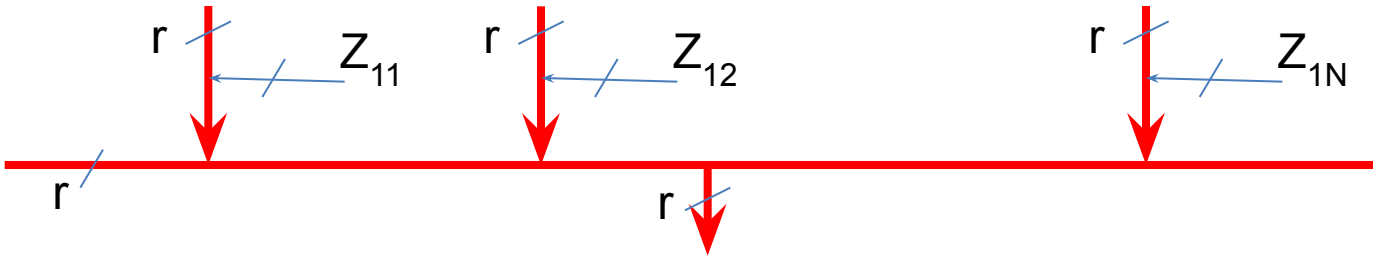
BUSES



EL MULTIPLEXOR PUEDE REEMPLAZARSE POR "TRISTATEs" Y UN BUS

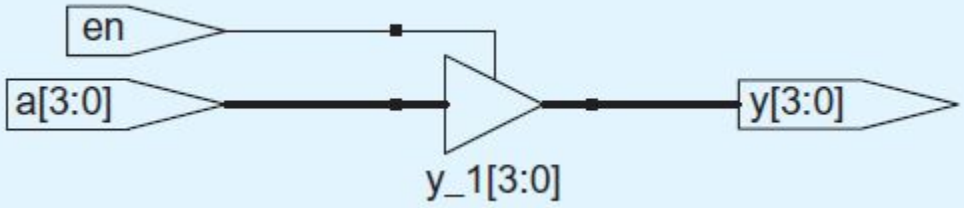


UNA FORMA SIMPLIFICADA DE REPRESENTAR EL BUS



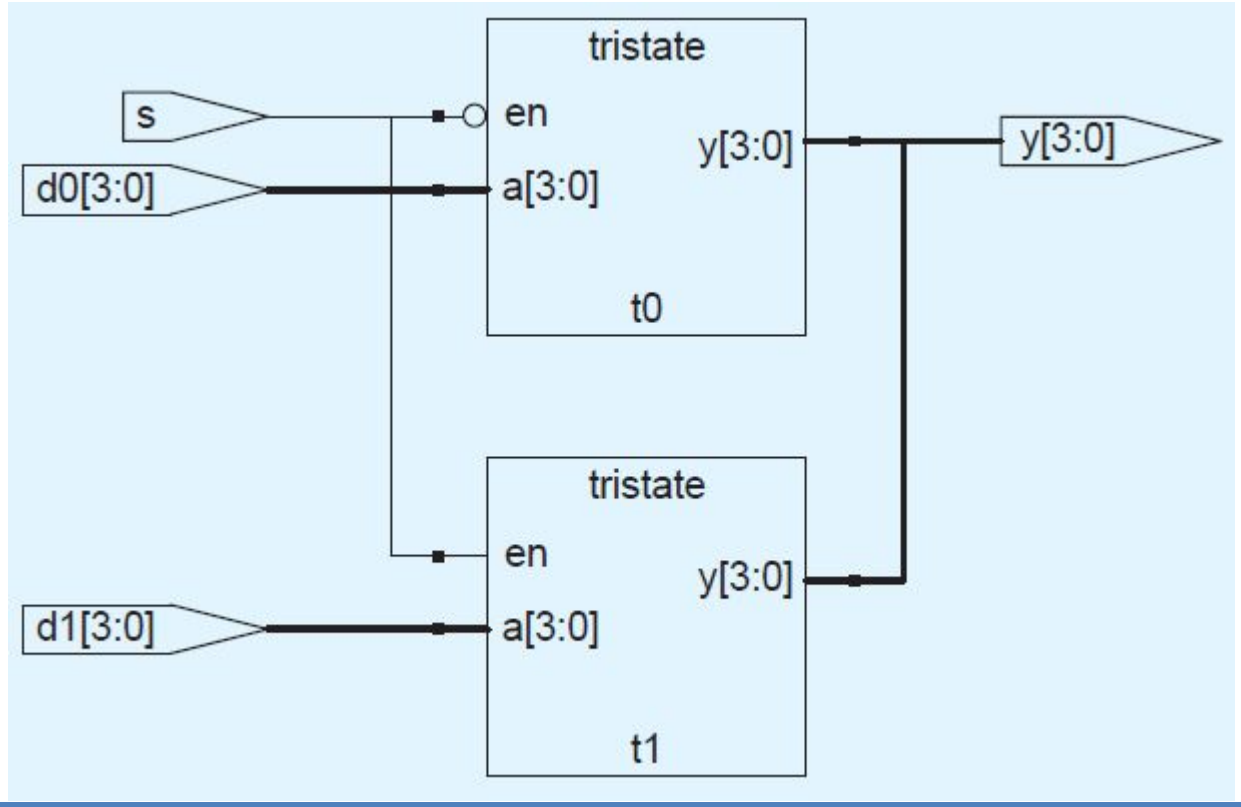
DESCIPCIÓN DEL MÓDULO TRISTATE

```
module tristate(input  [3:0] a,
                input        en,
                output [3:0] y);

  assign y = en ? a : 4'bz;
endmodule
```

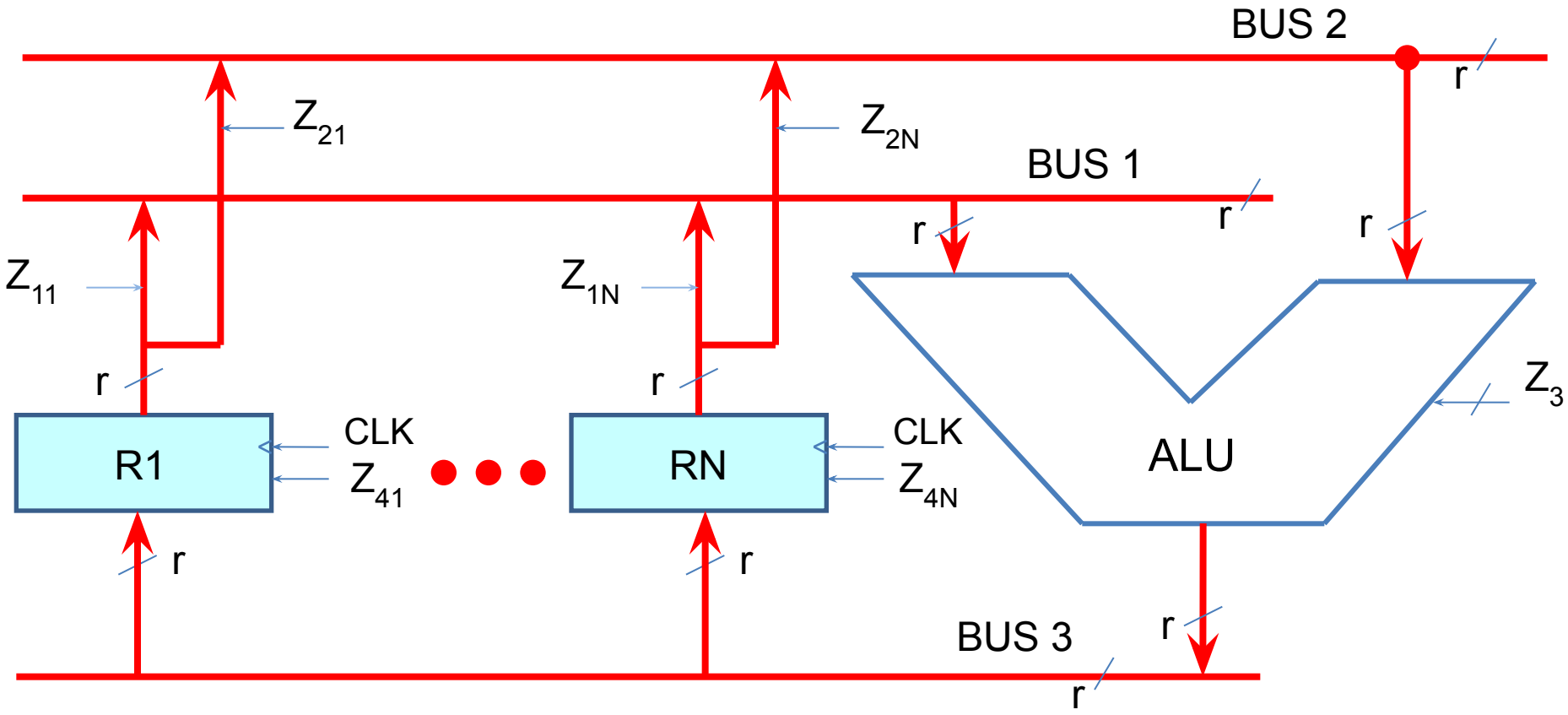


EJEMPLO UTILIZANDO TRISTATEs

```
module mux2 (input  [3:0] d0, d1,
             input        s,
             output [3:0] y);

  tristate t0 (d0, ~s, y);
  tristate t1 (d1, s, y);
endmodule
```



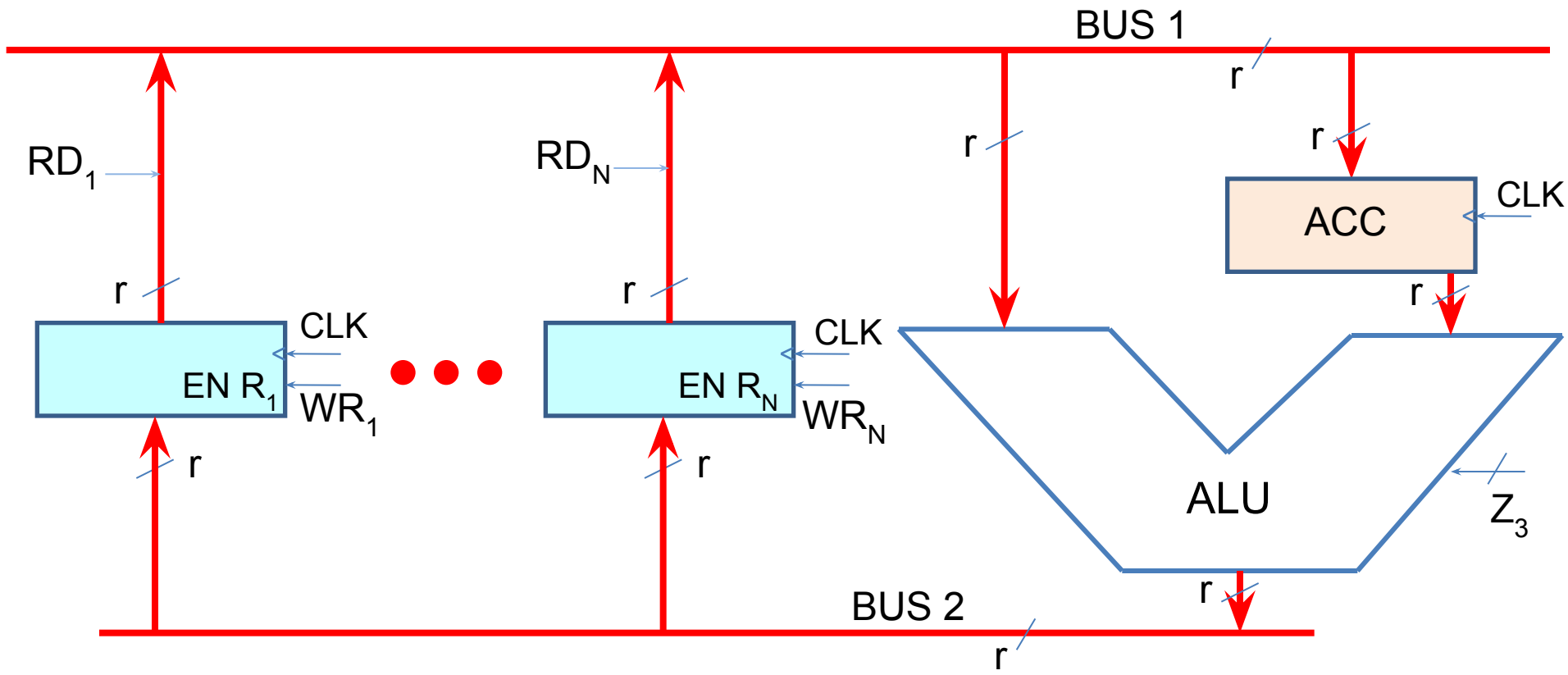
UNIDAD DE PROCESO DE 3 BUSES



$R_i := F(R_j, R_k)$

UTN

UNIDAD DE PROCESO DE DOS BUSES

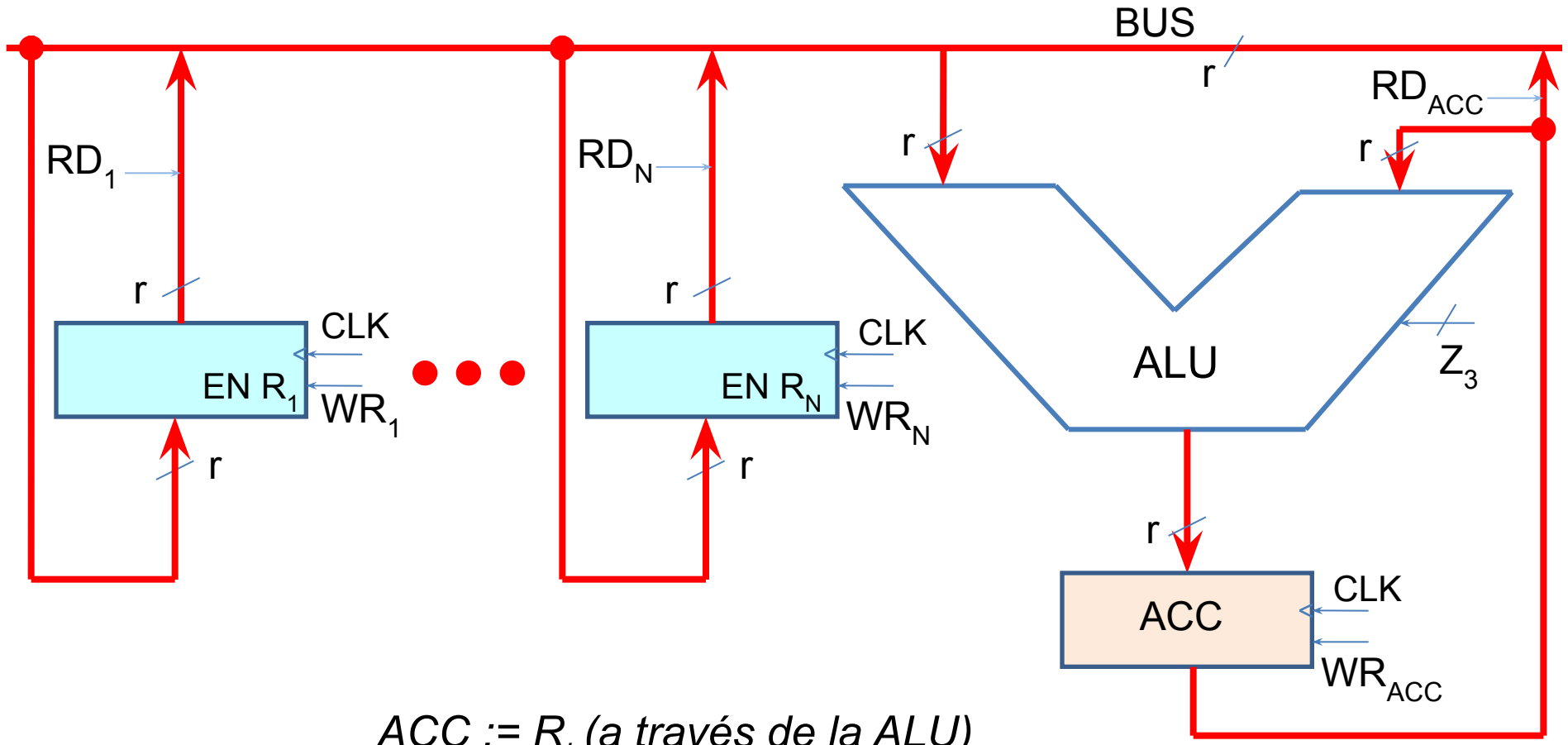


$R_i := F(R_j, R_k)$

$ACC := R_k$

$R_i := F(R_j, ACC)$

UNIDAD DE PROCESO DE UN BUS



$R_i := F(R_j, R_k)$

$ACC := R_j$ *(a través de la ALU)*

$ACC := F(ACC, R_k)$

$R_i := ACC$

EJEMPLO: MÁXIMO COMÚN DIVISOR (GREATEST COMMON DIVISOR)

El máximo común divisor de dos enteros distintos de cero es el mayor entero positivo que divide ambos números sin dejar resto.

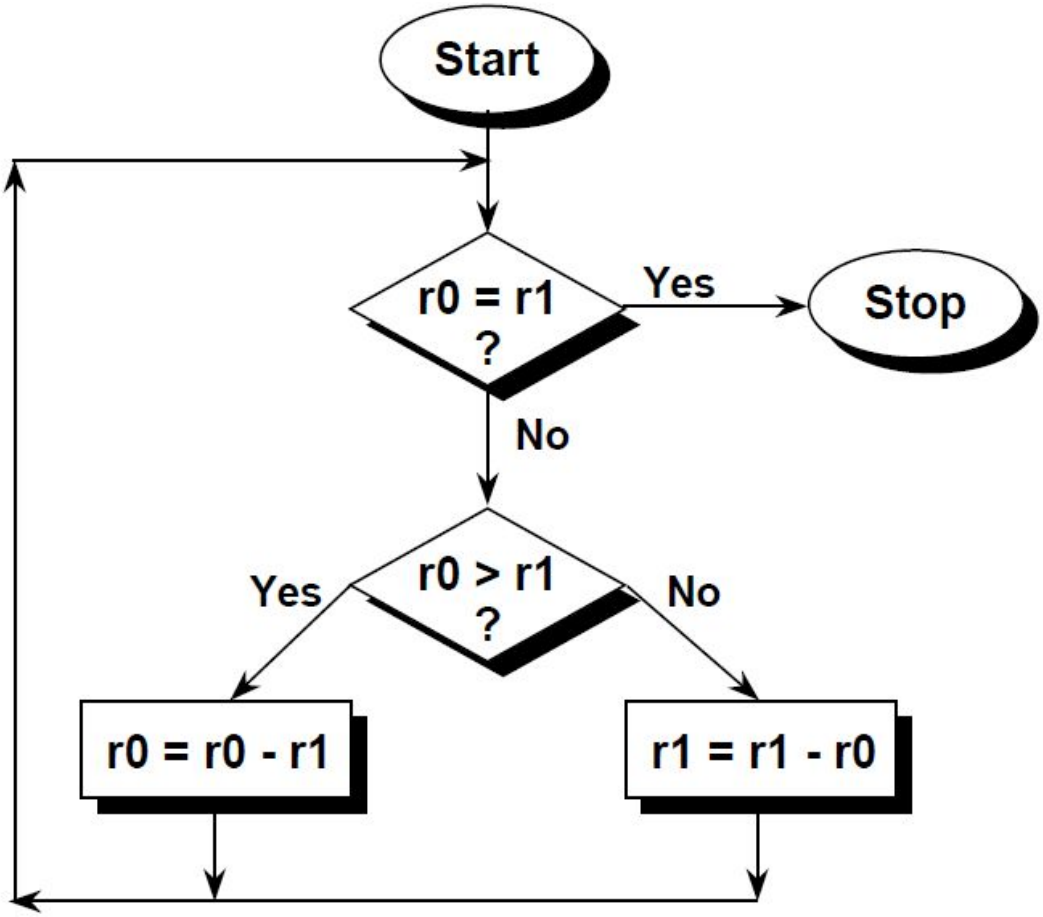
Estudie el siguiente pseudocódigo para asegurarse de comprender el algoritmo básico para calcular el MCD de dos números.

```
tmpX = X;
tmpY = Y;
while (tmpX != tmpY) {
        if (tmpX < tmpY)
              tmpY = tmpY-tmpX;
        else
              tmpX = tmpX-tmpY;
}
output = tmpX;
```

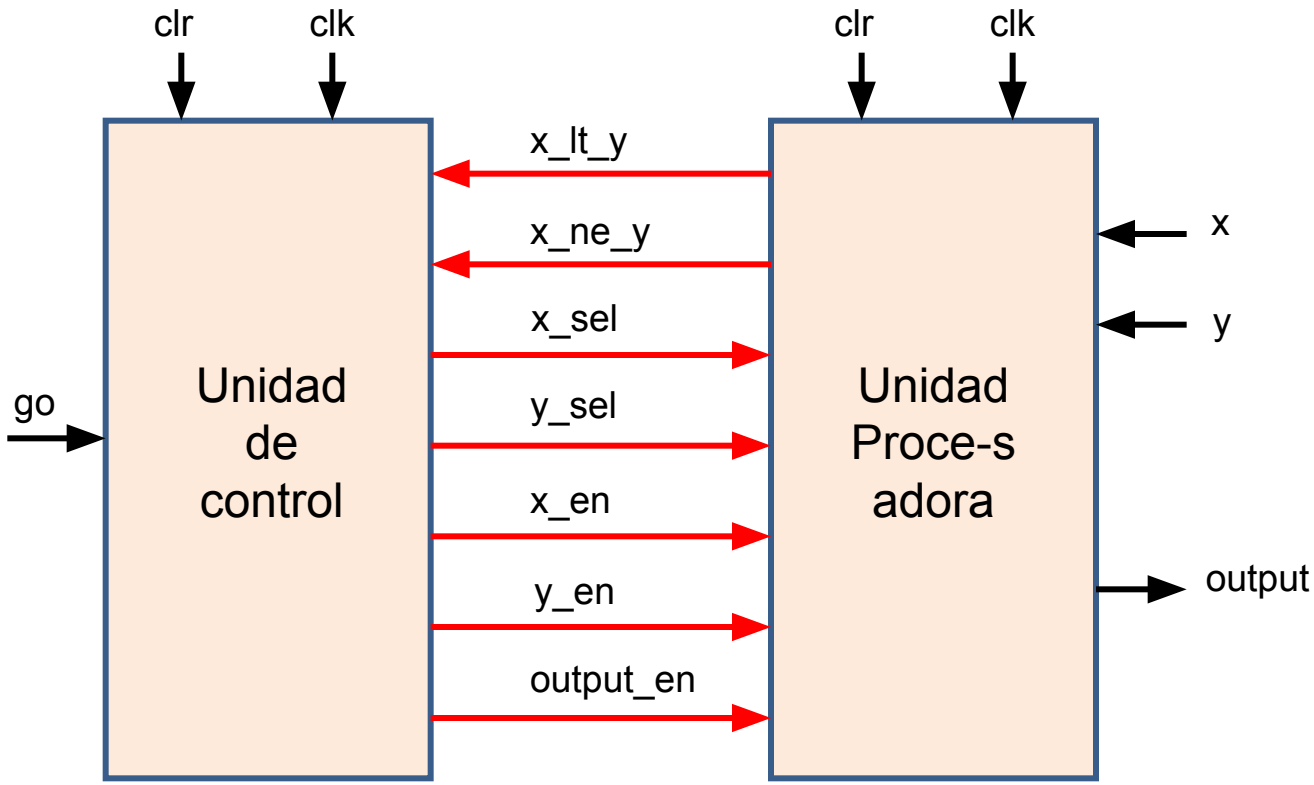
El código tiene 2 entradas (X e Y) y una salida (output).

```
int GCD( int inA, int inB)
{ int done = 0;
  int A = inA;
  int B = inB;
  while ( !done ) {
    if ( A < B )
    { swap = A;
      A = B;
      B = swap;
    }
    else if ( B != 0 )
      A = A - B;
    else
      done = 1;
  }
  return A;
}
```

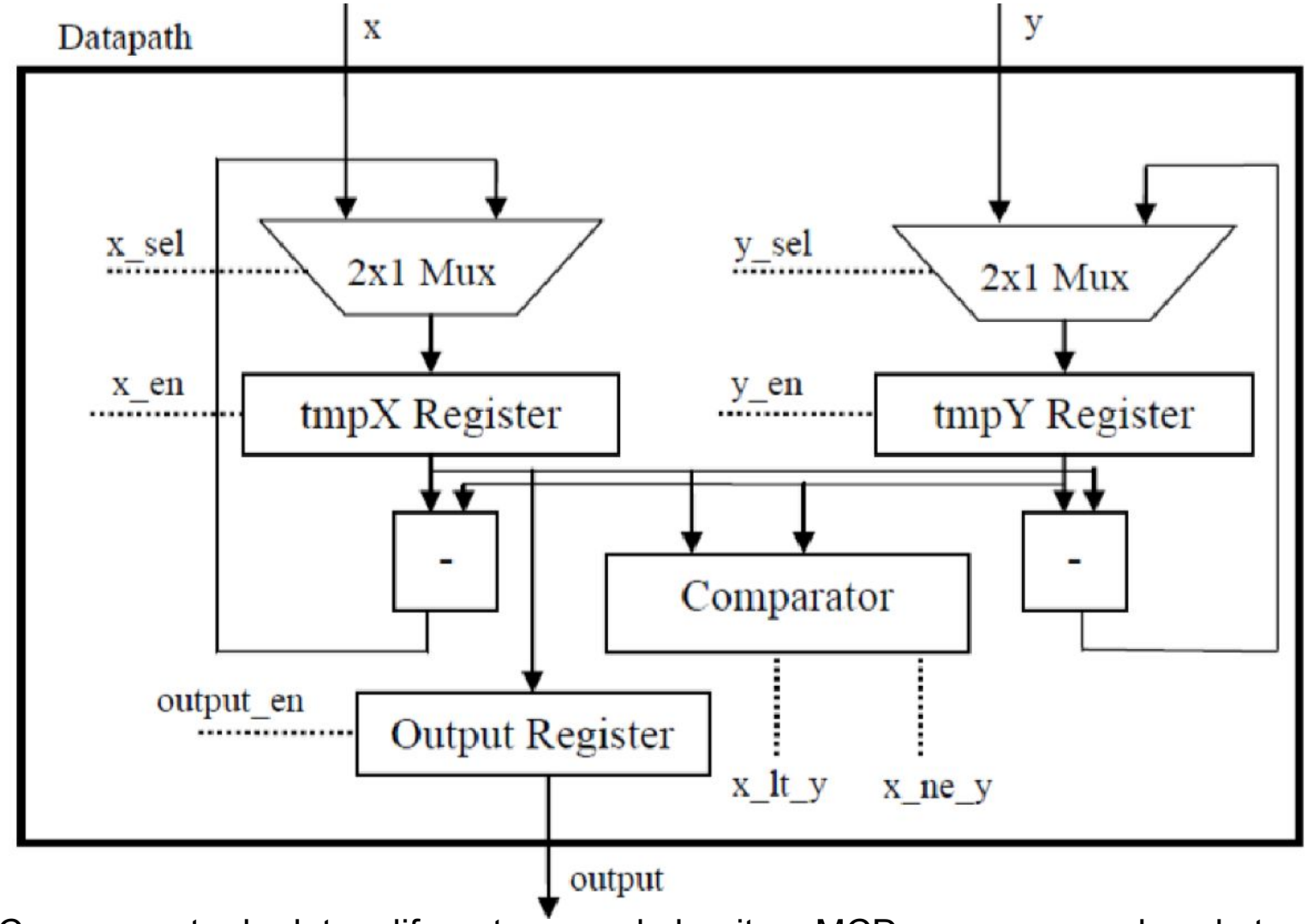
EJEMPLO: MÁXIMO COMÚN DIVISOR (GREATEST COMMON DIVISOR)



EJEMPLO: MÁXIMO COMÚN DIVISOR (DIAGRAMA EN BLOQUES)



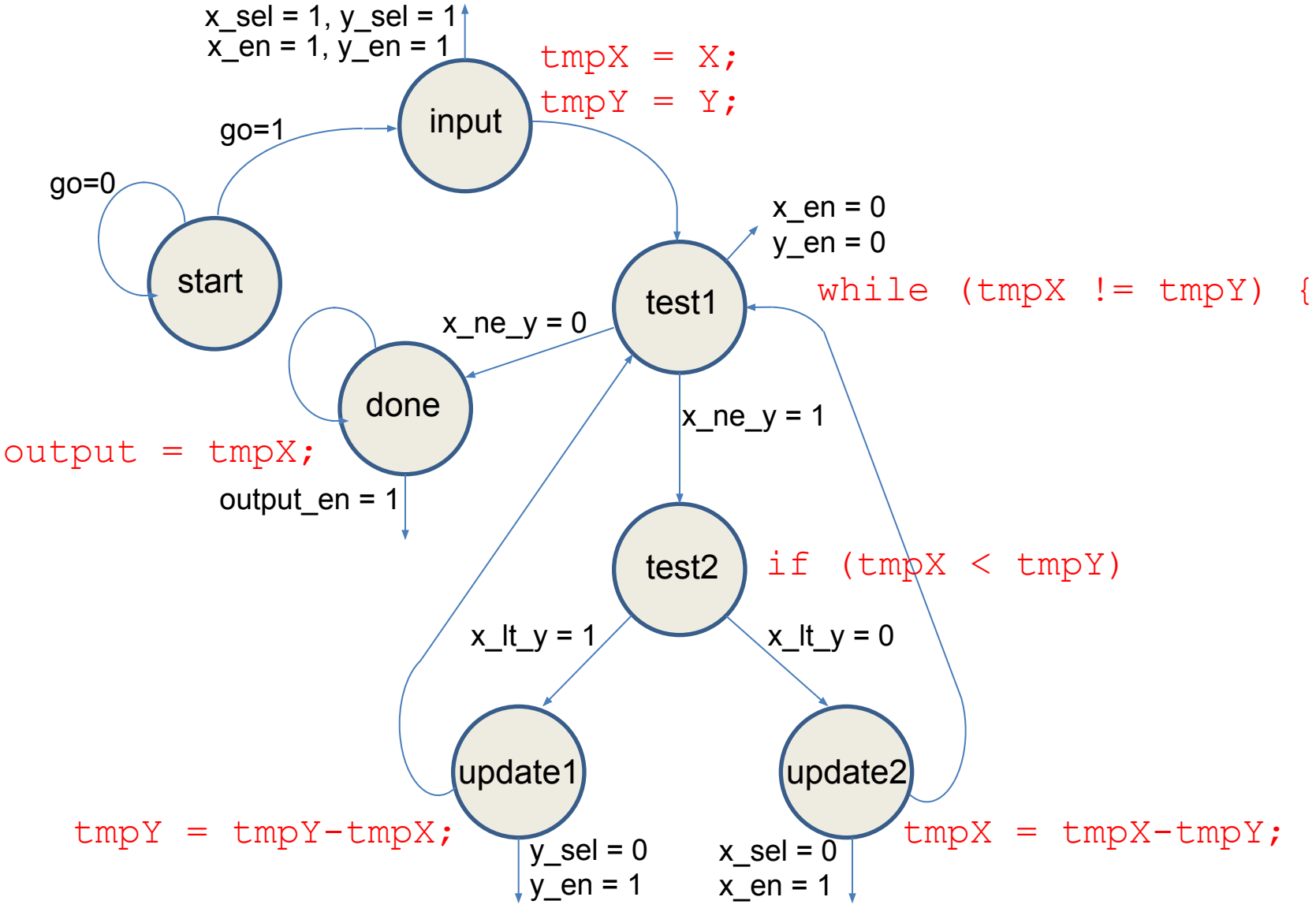
EJEMPLO: MÁXIMO COMÚN DIVISOR (UNIDAD PROCESADORA)



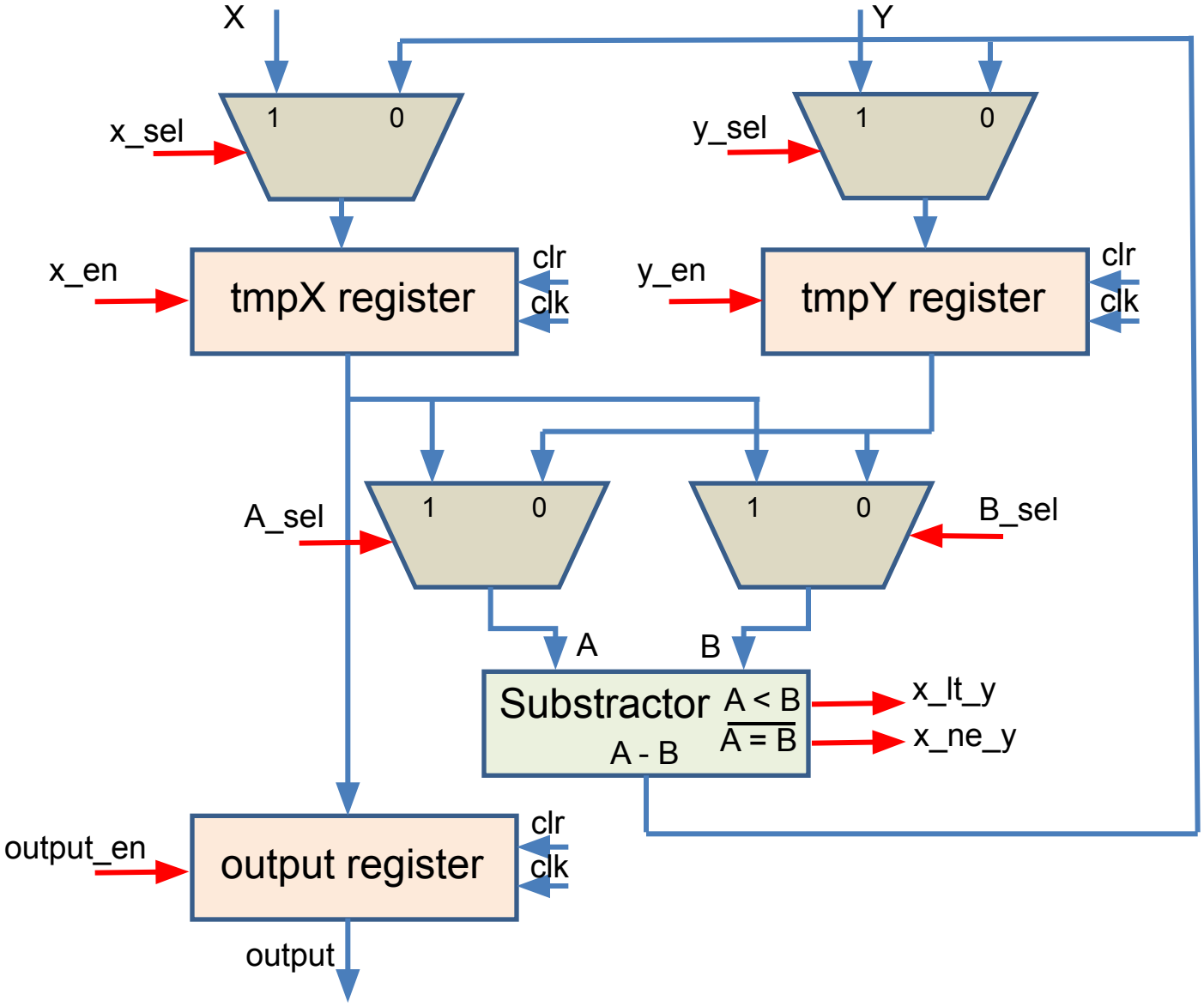
Todas las líneas punteadas representan entradas / salidas de control.

Cree una ruta de datos diferente para el algoritmo MCD que use un solo substractor. Agregue todos los componentes y/o señales de control que sean necesarios.

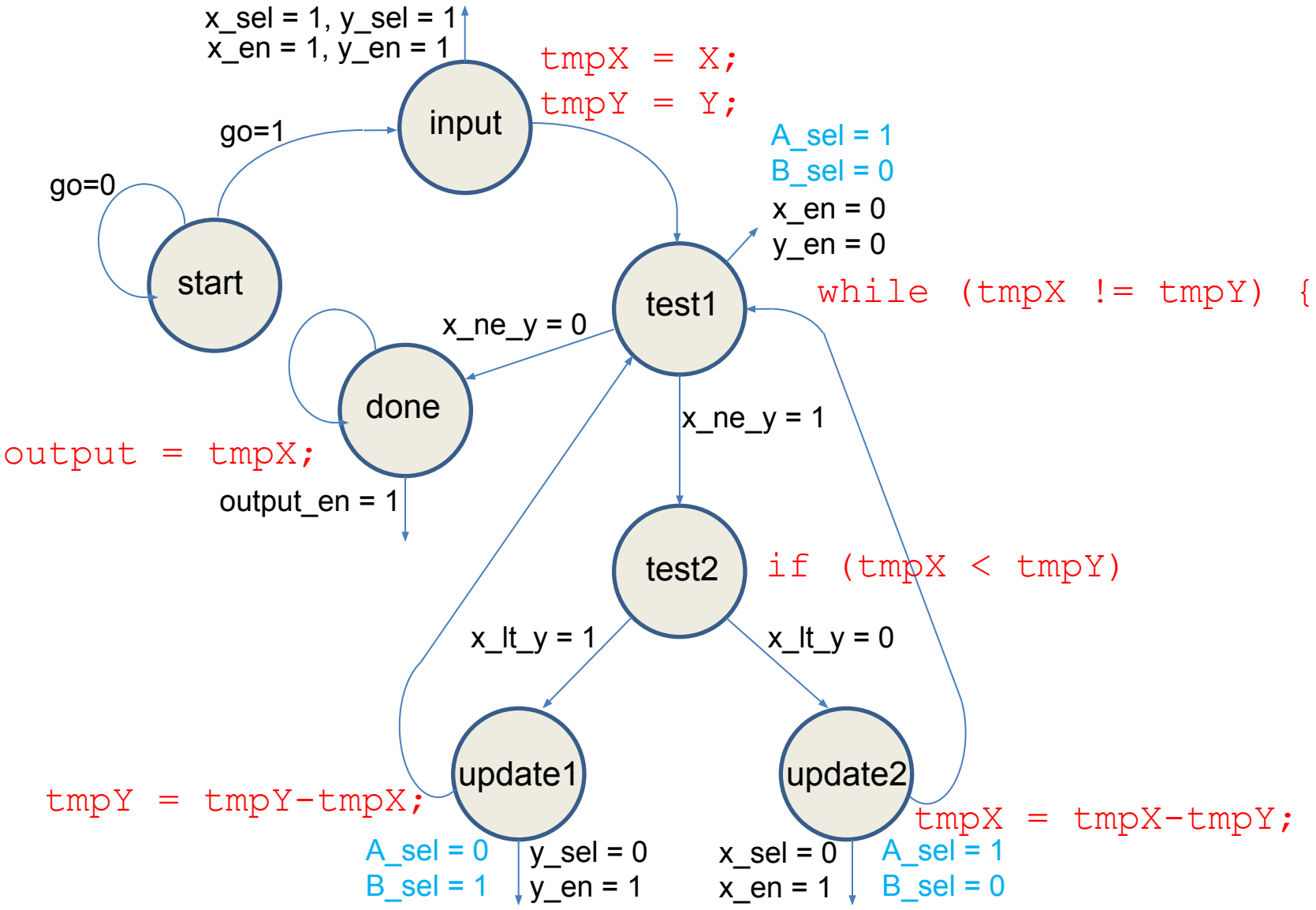
EJEMPLO: MÁXIMO COMÚN DIVISOR (UNIDAD DE CONTROL)



EJEMPLO: MÁXIMO COMÚN DIVISOR (UNIDAD PROCESADORA OPTIMIZADA)



EJEMPLO: MÁXIMO COMÚN DIVISOR (UNIDAD DE CONTROL)



PASOS PARA CREAR LA UNIDAD PROCESADORA (DATAPATH)

1) Dibuje un registro (cuadro rectangular con entrada en la parte superior y salida en la parte inferior) para cada variable presente en el algoritmo. Para el algoritmo GCD, estas incluirían x, y, y output (gcd).

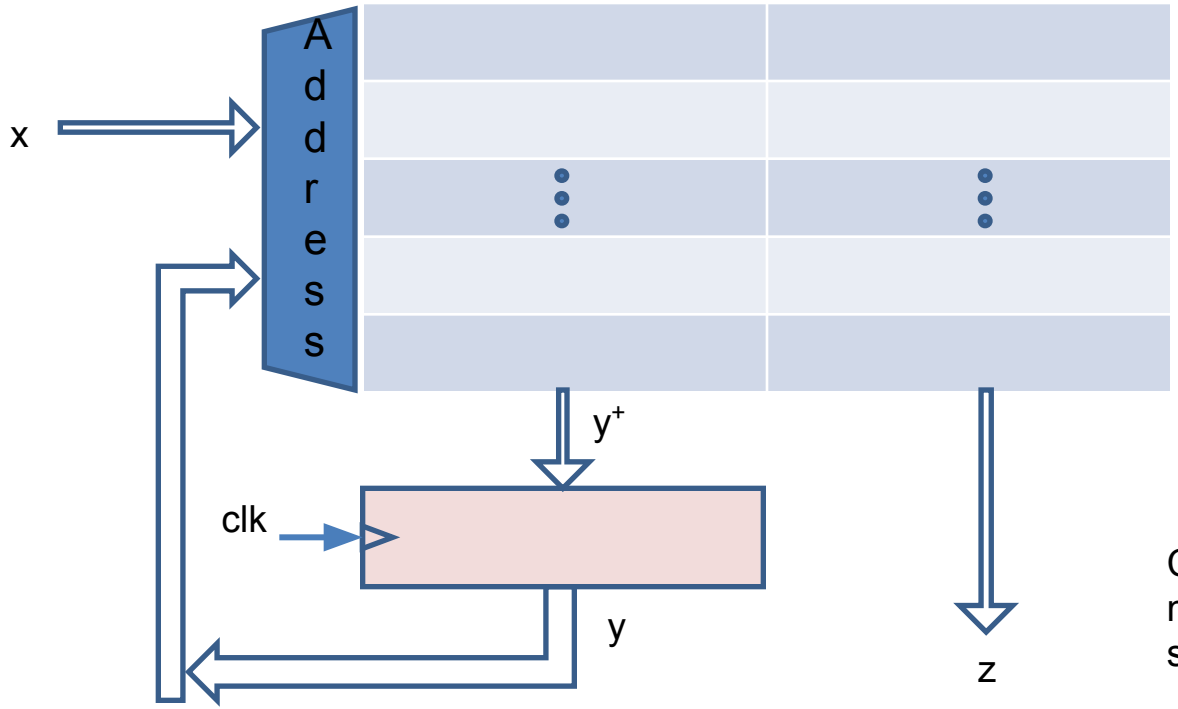
2) Definir bloques combinacionales para implementar cualquier operación aritmética o lógica necesaria.

3) Conecte las salidas de los registros a las entradas de las operaciones aritméticas y lógicas apropiadas, y conecte las salidas de las operaciones aritméticas y lógicas a los registros apropiados. Los multiplexores se pueden usar si la entrada a un registro puede provenir de más de una fuente.

UTN

DIFERENTES TIPOS DE REALIZACIÓN DE LA UNIDAD DE CONTROL

La unidad de control de una máquina algorítmica es una máquina secuencial cuyas entradas "*x*" son *variables de condición*, es decir, variables procedentes de la unidad de proceso y dando información sobre el estado de la misma, y cuya salidas "*z*" son *variables de control*, es decir, variables transmitidas a la unidad de proceso y controlando el funcionamiento de la misma (buses, ALU, registros, etc.).



Un circuito combinacional realizado por medio de una memoria ROM

Un registro almacenando el número del estado presente de la máquina.

Cada fila de la ROM consta de dos campos:
-El número del estado siguiente.
-El valor de las variables de salida (variables de control z).

Cuando se utiliza este tipo de materialización, el contenido de la ROM suele llamarse *microprograma*.

A cada estado de la máquina corresponde una *microinstrucción* cuya dirección viene dada por el contenido del registro. A una microinstrucción corresponden tantas filas de la ROM como posibles combinaciones de las variables de entrada x.

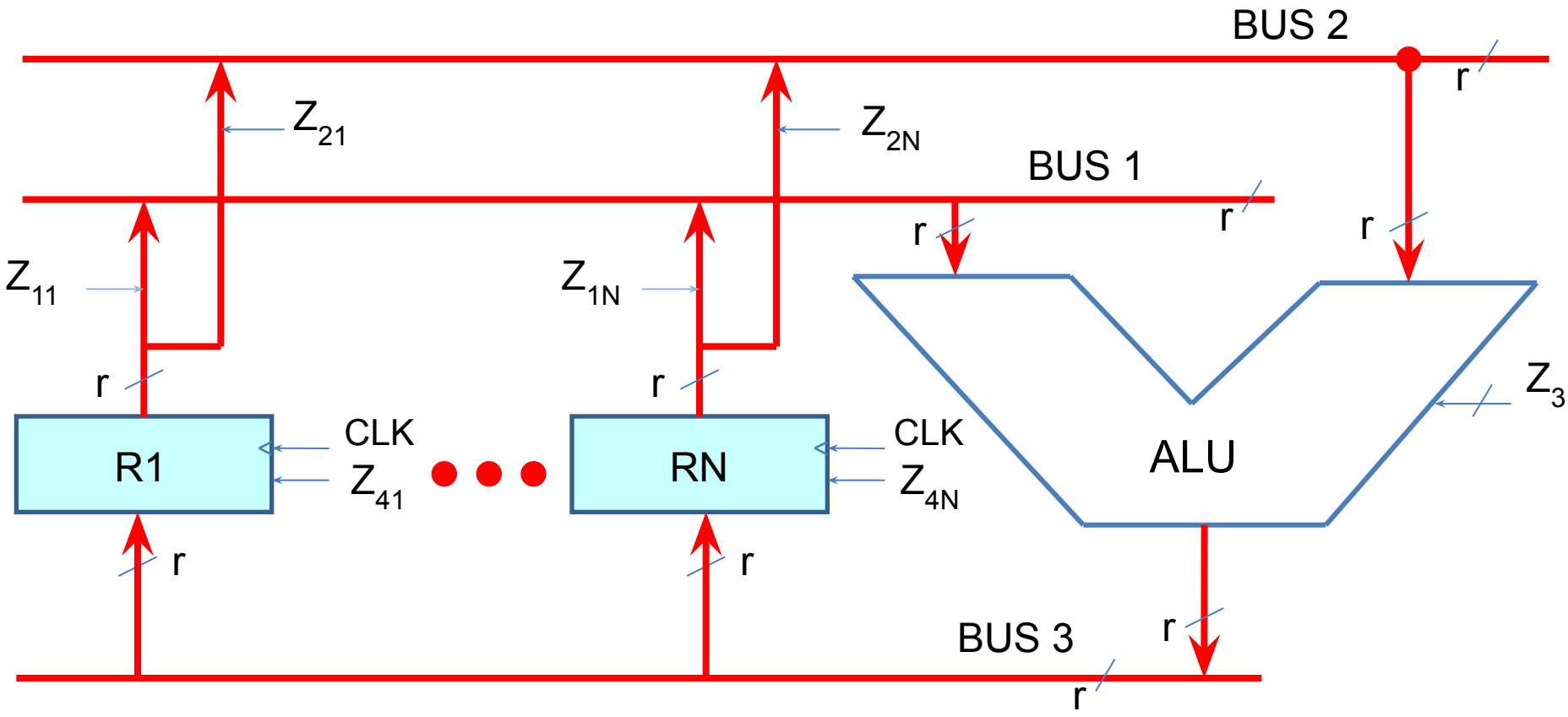
DIFERENTES TIPOS DE REALIZACIÓN DE LA UNIDAD DE CONTROL

A la primera máquina secuencial del máximo común divisor corresponde el siguiente contenido de la ROM

Dirección de la ROM						Contenido de la ROM							
y_2	y_1	y_0	go	x_lt_y	x_ne_y	y_2^+	y_1^+	y_0^+	x_sel	y_sel	x_en	y_en	output_en
0	0	0	0	x	x	0	0	0	0	0	0	0	0
0	0	0	1	x	x	0	0	1	0	0	0	0	0
0	0	1	x	x	x	0	1	0	1	1	1	1	0
0	1	0	x	x	0	1	1	0	x	x	0	0	0
0	1	0	x	x	1	0	1	1	x	x	0	0	0
0	1	1	x	0	x	1	0	0	x	x	0	0	0
0	1	1	x	1	x	1	0	1	x	x	0	0	0
1	0	0	x	x	x	0	1	0	0	x	1	0	0
1	0	1	x	x	x	0	1	0	x	0	0	1	0
1	1	0	x	x	x	1	1	0	x	x	0	x	1

UTN

REALIZACIÓN DE LA UNIDAD DE CONTROL: CODIFICADOR DE ÓRDENES

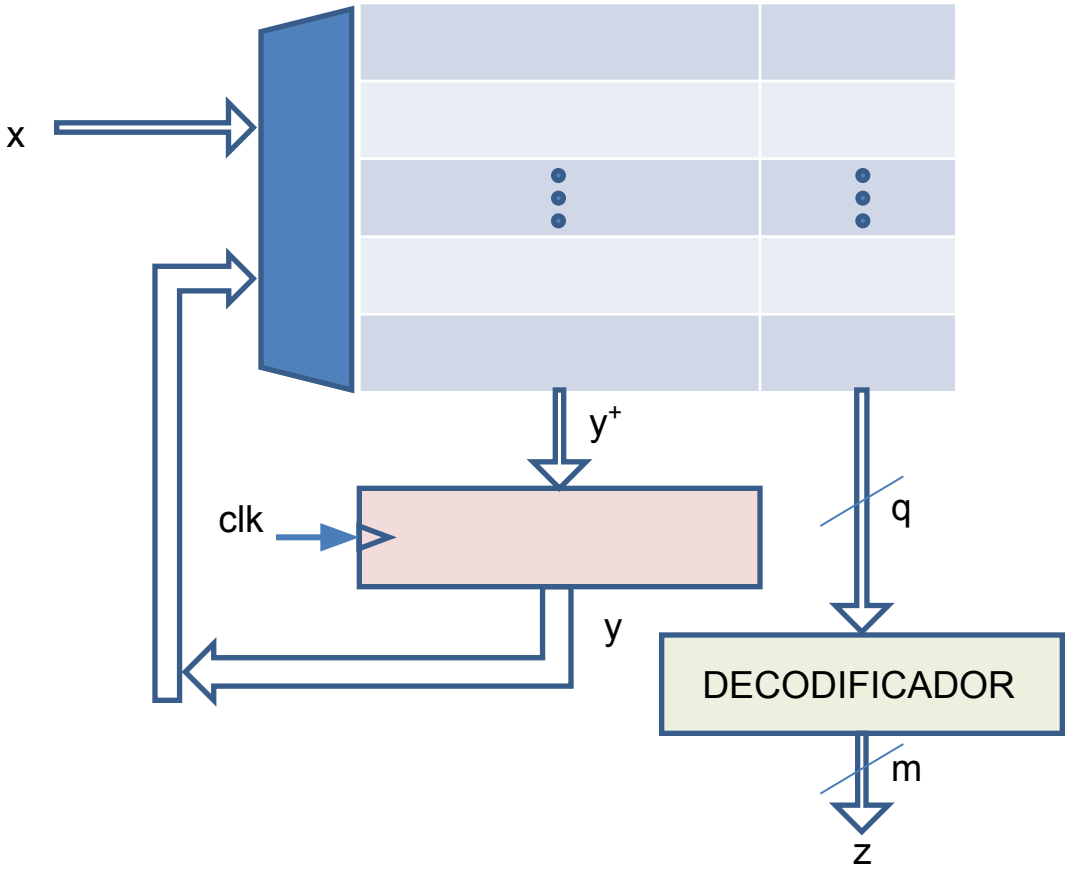


La unidad de control asociada a la unidad de proceso de la figura genera 4 grupos de variables de control:

1)z11,z12,…,z1N
2)z21,z22,…,z2N
3)z3
4)z41,z42,…,z4N

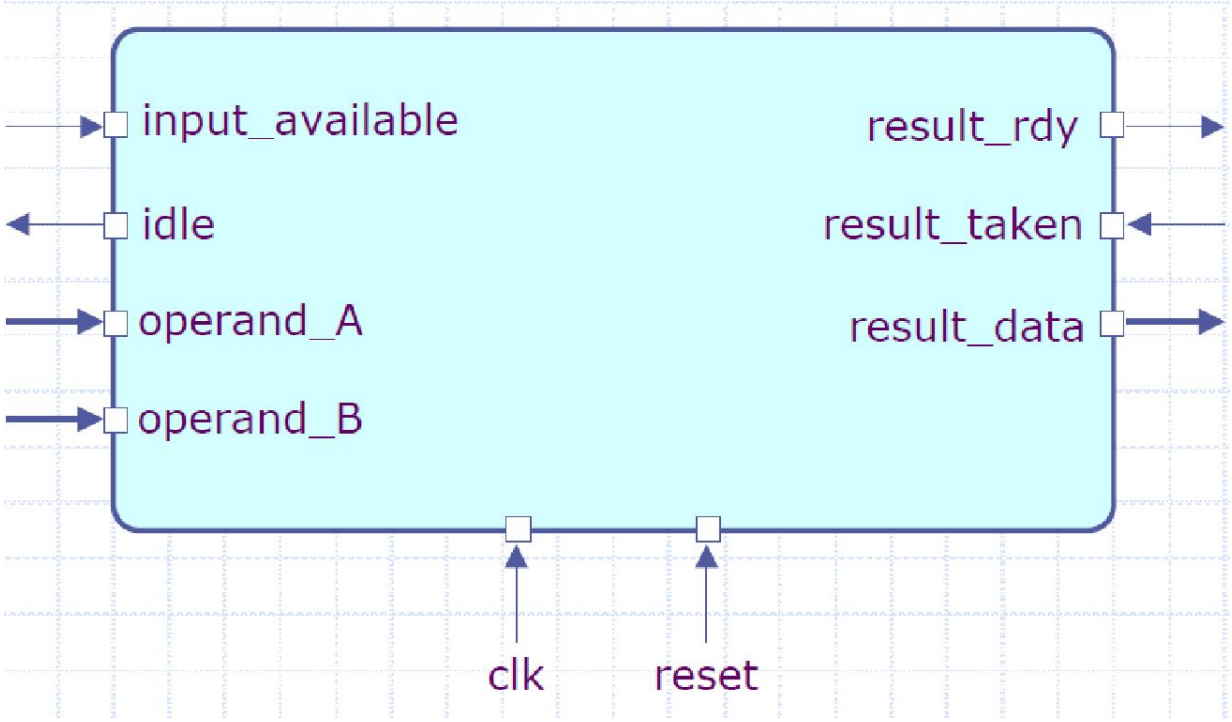
REALIZACIÓN DE LA UNIDAD DE CONTROL: CODIFICADOR DE ÓRDENES

Es probable que el número de vectores z diferentes sea mucho menor que 2^q. Por lo tanto, la misma información puede almacenarse en la memoria en forma codificada con menos bits, por ejemplo, con q bits. Un circuito decodificador genera z a partir de las órdenes en forma codificada.



EJEMPLO: MÁXIMO COMÚN DIVISOR VERSIÓN II

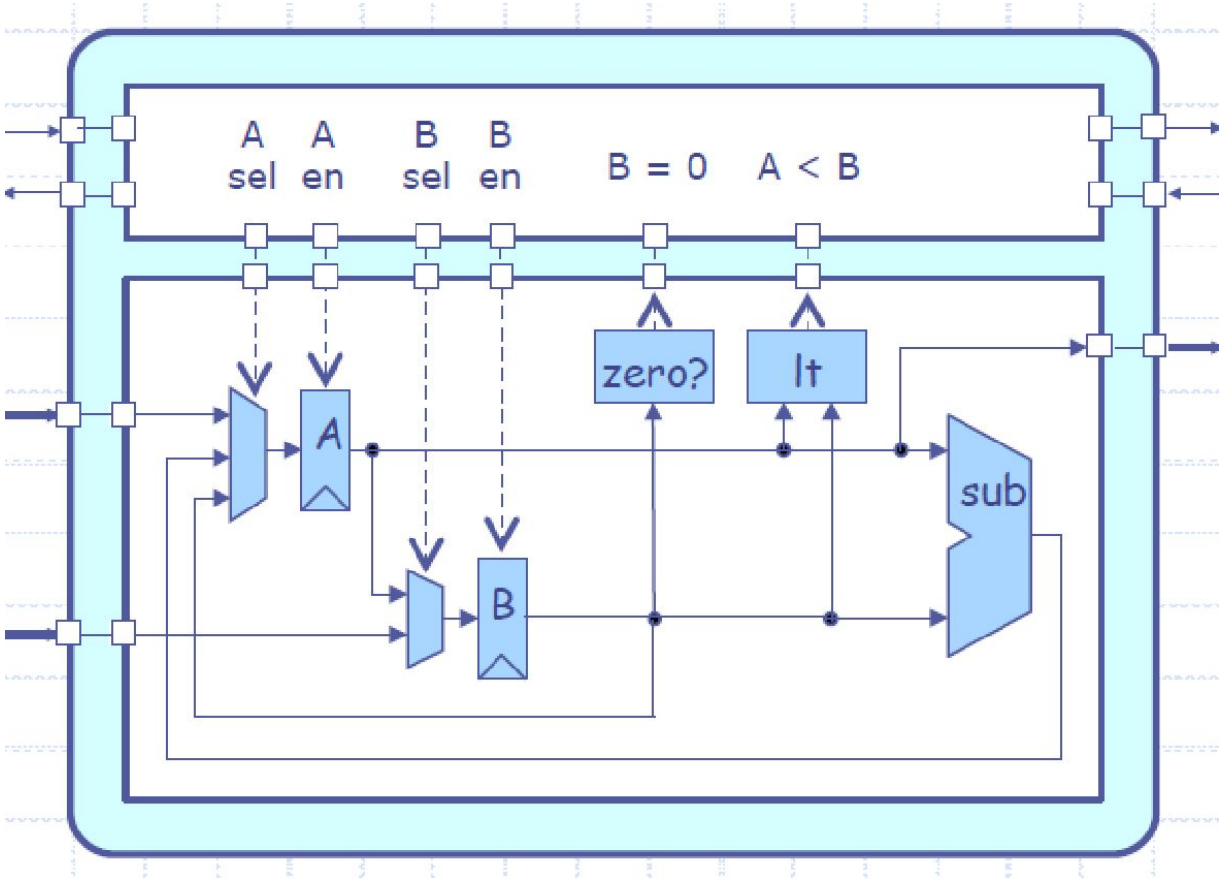
Diseñar una interfaz de puerto adecuada




```
int GCD( int inA, int inB)
{ int done = 0;
  int A = inA;
  int B = inB;
  while ( !done ) {
    if ( A < B )
    { swap = A;
      A = B;
      B = swap;
    }
    else if ( B != 0 )
      A = A - B;
    else
      done = 1;
  }
  return A;
}
```

EJEMPLO: MÁXIMO COMÚN DIVISOR VERSIÓN II

Diseñar una ruta de datos (datapath) que tenga las unidades funcionales

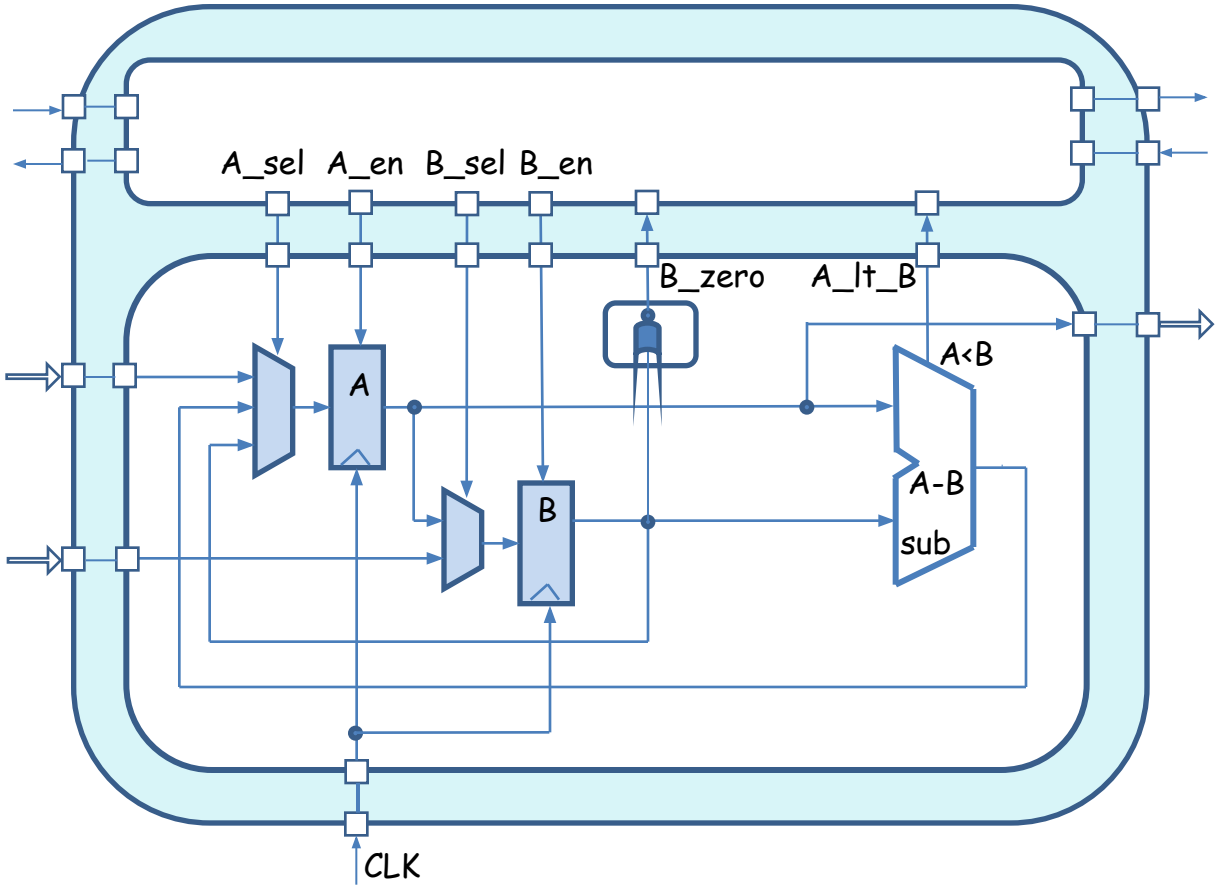



```
int GCD( int inA, int inB)
{ int done = 0;
  int A = inA;
  int B = inB;
  while ( !done ) {
    if ( A < B )
    { swap = A;
      A = B;
      B = swap;
    }
    else if ( B != 0 )
      A = A - B;
    else
      done = 1;
  }
  return A;
}
```

UTN

EJEMPLO: MÁXIMO COMÚN DIVISOR VERSIÓN II

La unidad de control debe estar diseñada para: estar ocupada o esperando entrada o esperando que la salida sea tomada.




```
int GCD( int inA, int inB)
{ int done = 0;
  int A = inA;
  int B = inB;
  while ( !done ) {
    if ( A < B )
    { swap = A;
      A = B;
      B = swap;
    }
    else if ( B != 0 )
      A = A - B;
    else
      done = 1;
  }
  return A;
}
```

MÁXIMO COMÚN DIVISOR (VERSIÓN II)
UNIDAD DE CONTROL

La unidad de control requiere una máquina de estado para señales válidas / listas