**HERRAMIENTAS UTILIZADAS EN EL DISEÑO DE PLATAFORMAS DE HARDWARE EN L.T.R.**

1. **EL ALU: (Artimetic logic unit)**

Es un bloque digital combinacional de alta escala de integración, diseñado con el objetivo de centralizar en él, todas las necesidades de ejecución de operación lógicas y aritméticas que el sistema en el cual el este inserto, necesita desarrollar. Estos bloques poseen 4 tipos de líneas digitales asociadas, en figura de abajo se pueden observar.

En el diseño de un sistema digital implementado por L.T.R. el ALU es el bloque digital más importante, de hecho en un CPU al ALU se le llama el “corazón del CPU”

Entradas de control de Entradas de Concatenacion

ejecución de operaciones proveniente de un ALU

“anterior” en conexión

cascada

Flujo de información en ambos OPERANDOS (1 &2) de “n” bits

ALU

OP1 0P2

OP1 xor OP2 Cin

OP1 and OP2 B in

OP1 + OP2 OP1 < 0P2

.. ..

.. ..

Cout

Bout

OP1 < OP2 R

..

..

.

Salida de información computada de “n” bits

Salidas para

concatenación para conexión

a un ALU “siguiente” en cascada

1. **BUS COMUN:**

En electrónica digital, la palabra “bus” evidencia un conjunto de líneas digitales que portan genéricamente el mismo tipo de información

Bus común, involucra mantener un solo tamaño de líneas para mantener comunicación entre dos o más sistemas digitales como fuente hacia un solo sistema digital como destino, cuando realmente deberían existir varios buses ( uno por cada sistema digital fuente) en lugar de uno solo

La figura mostrada abajo evidencia de donde nace el concepto de “bus común”. Note que uno de los grandes obstáculos para poder conectar dos o más sistemas digitales fuente, hacia un solo sistema digital destino radica en el conflicto óhmico (corto circuito) que se da cuando conectamos juntas dos líneas físicas diferentes con posibilidad de tener diferente valor de tensión (voltaje) entre ellas

NECESIDAD:

SISTEMA

DIGITAL

FUENTE 1

SISTEMA

DIGITAL

DESTINO

SISTEMA

DIGITAL

FUENTE 2

PELIGRO DE CONFLICTO OHMICO (corto circuito)

SOLUCION:

Existen 3 soluciones básicas: 1) utilizar multiplexadores 2) utilizar elementos de alta impedancia “Z” 3) utilizar lógica de “colector común” (esta última no la evaluaremos)

SOLUCION POR MULTIPLEXADORES:

SISTEMA

DIGITAL

CANAL 0

FUENTE 1

SISTEMA

MUX DIGITAL

2 X 1 Y

DESTINO

SISTEMA

DIGITAL

CANAL 1

FUENTE 2

So A este bus se le llama “BUS COMUN”

Pues logro agrupar en uno solo la

Información que llega a un sistema digital

“destino” proveniente de 2 o más “fuentes”

Selector de paso

SOLUCION POR ALTA IMPEDANCIA (TRI-STATE):

SISTEMA BUS COMUN

DIGITAL

FUENTE 1 SISTEMA

DIGITAL

Z´

decoder Yo

2 X 1

I Y1 DESTINO

SISTEMA

DIGITAL

selector FUENTE 2

Z´

DONDE: “Z” son entradas digitales de control

De alta impedancia

NOTA:

* Los bloques digitales FUENTE con alta impedancia funcionan asi:

Si Z = 0 entonces ellos tienen salidas validas

Si Z = 1 entonces ellos no tienen valores lógicos valederos a la salida, “virtualmente es como si ellos hubiesen sido extraidos del protoboard o fibra de vidrio

* El decoder que los maneja, trabaja en lógica negativa, y por tanto solo mantiene activo a un bloque digital a un tiempo, asi NO EXISTE CONFLICTO

1. **RETARDOS: (DELAY)**

El retardo es una técnica utilizada para equalizar la velocidad de respuesta de dos sistemas digitales que poseen diferente velocidad de respuesta (lectura o escritura en ellos) y que desean comunicarse óptimamente.

Existen 3 tipos de Retardos:

* Retardos por hardware
* Retardos por software
* Retardos por firmware (i.c.´s programables)

3.1) Retardos por hardware:

3.1.1) Utilizando el tiempo de propagación por compuerta [ t (tp) ]:

f(t) f (t) SIN retardo

**O O** f [t + 2( t (tp)) ]

Donde: t (tp) es el tiempo de propagación de cada uno de los inversores

Decimos por tanto que: la señal digital f [ t + 2 ( t (tp) ) ] llego mas tarde que la señal digital

Que viajo por el conductor y que llego sin retardo f (t)

3.1.2 ) Utilizando el tiempo de carga exponencial de un circuito a base de “resistor-capacitor” o

“R-C”

f (t) f (t)

R f [ t + 3RC ]

C

Donde: decimos que la señal digital dependiente del tiempo f(t) que viajo a través de la

Red “R-C” llego retardada respecto de la que viajo por el conductor, un diferencia

De tiempo dada por 3 RC. Donde R esta en ohmios y C esta en faradios

3.2 ) BLOQUE UTILIZADO PARA EL DELAY EN L.T.R.

Entrada f(t) D Salida retardada de f(t)

1. **CONTADOR JOHNSON:**

El contador Johnson esta compuesto por un contador binario de “k” bits ascendente, cíclico, seguido de un decodificador de línea de k X 2 exp (k) en lógica positiva.

El objetivo de este contador es de proporcionarle a un sistema digital donde esta inmerso, de la perfecta sincronización y temporización de las diferentes líneas digitales involucradas en la inducción algorítmica que el sistema digital se propone desarrollar para alcanzar un objetivo

Comúnmente un resultado final, producto de la ejecución secuenciada de una serie de sentencias en L.T.R.

4.1) ESQUEMA DIGITAL DE UN CONTADOR JOHNSON DE 4 LINEAS DE TIEMPO

T3 T2 T1 T0 4 lineas de tiempo

De salida

D C B A

DECODER DE 2X4

MSB LSB

MSB LSB

RELOJ con f (Hertz)

Contador binario de 2 bits

4.2) VISTA A BLOQUE DE UN CONTADOR JOHNSON

CONTADOR

JOHNSON

T0

T1

RELOJ de f (hertz)

T2

T3

4.3) DIAGRAMA DE TIEMPO

RELOJ

O t (seg)

T0 t (seg)

0

T1 t (seg)

0

T2 t (seg)

O

T3 t (seg)

O

1. **REGISTROS TEMPORALES Y REGISTROS DE USOS GENERALES**

Ya tenemos definiciones de registros, en esta sección veremos cuando se usan registros temporales y el diagrama a bloque que en algoritmos de L.T.R. se usan para especificar a un registro en general

La diferencia radical entre un registro cualquiera y uno temporal, es que: el registro cualquiera es especificado comúnmente en la sintaxis de los comando L.T.R. Mientras el registro temporal, no aparece en la estructura algorítmica de L.T.R. que genera el hardware pedido y es el diseñador el que a criterio lo coloca donde le sirve, previo discernimiento

5.1) BLOQUE REPRESENTATIVO DE UN REGISTRO, CON LINEAS DE CONTROL PARA LECTURA Y ESCRITURA

BUS CON INFORMACION ENTRANDO AL REGISTRO “MBR”

línea de control de

escritura

**LOAD´ MBR**

**Z´**

Línea de alta impedancia

Para lectura protegida

BUS CON INFORMACION SALIENDO DEL REGISTRO “MBR”

**PROCEDIMIENTO DE DISEÑO CON L.T.R.**

Existen dos ramas que podemos explotar en el conocimiento de estructuras en hardware en L.T.R. y los algoritmos que las generan (diseño) o los algoritmos en L.T.R. que se obtienen producto del análisis de las plataformas de hardware ya diseñadas (análisis)

Nosotros veremos “DISEÑO”

PASOS:

1. Elegir lo bloques a utilizar
2. Ubicarlos estratégicamente, esto es, los mejor ubicados para que las líneas de flujo de información entre ellos y sus líneas de control no generen desorden
3. Generar e interconectar las líneas de flujo de información entre los bloques
4. Interconectar las líneas de control entre los bloques y el contador Johnson, avaladas por habilidad ganada con el curso o utilizar las herramientas de diseño aprendidas a través del curso

Ejemplo de diseño con L.T.R.:

Elegir e interconectar adecuadamente la plataforma de hardware que satisfaga la ejecución del siguiente algoritmo en L.T.R.

LINEA 1: T1: A 🡨 B xor C

LINEA 2: T2: D 🡨 E and F

Donde:

* T1 & T2 son señales provenientes de un contador Johnson
* A,B,C,D,E,F son registros de “n” bits
* Las micro operaciones lógicas “xor” & “and” deben ejecutarse en un mismo ALU

B E C F

CH0 CH1 CH0 CH1

So MUX 2X1 So MUX 2X1

Y y

OP1 0P2

CONTADOR OP1 xor OP2

JOHNSON

. OP1 and OP2 ALU

To

T1

R

T2

D

A D

RELOJ

D