

Tema 2 Segmentacion y Replicacion

Tema 2 Segmentacion y Replicacion

Diseños Serie

Segmentación

Elementos de Desacoplo

Celda D (latch D)

Registro D (Flip-flop D)

Tipos de U.F. Segmentadas

Interpretación de instrucciones Serie

Segmentación del proceso de interpretación de instrucciones

Conflicto instrucciones con diferente | mismo patrón

Control de un dispositivo segmentado

Control de iniciaciones

Control lógico

Diseños Serie

En un diseño serie hay que esperar a que finalice una tarea para iniciar una nueva. La productividad es la inversa del tiempo de ciclo, es decir, el tiempo en realizar una operación.

Segmentación

La segmentación permite diseñar un sistema donde se empiece a procesar una operación antes de acabar la actual. Debemos distinguir las diferentes fases, los diferentes módulos combinacionales de cada fase y establecer un secuenciamiento adecuado.

Entre las diferentes etapas, colocaremos registros de desacoplo para que los cálculos de cada etapa no afecten a la etapa siguiente hasta el flanco del reloj. La latencia de una tarea será el tiempo total de procesamiento de la tarea, en ciclos de reloj.

El tiempo de ciclo, debe garantizar que las lógicas de las diferentes etapas disponen del suficiente tiempo para procesar los datos de entrada:

$$T_c \geq \max(T_{E_i}) + T_p$$

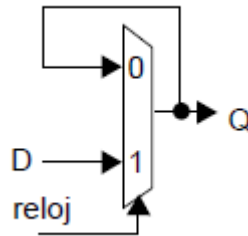
La ganancia máxima ideal será el número de etapas.

$$T_{seg} = N(T_p + T_L/NE) + (NE - 1) * (T_p + T_L/NE)$$

Elementos de Desacoplo

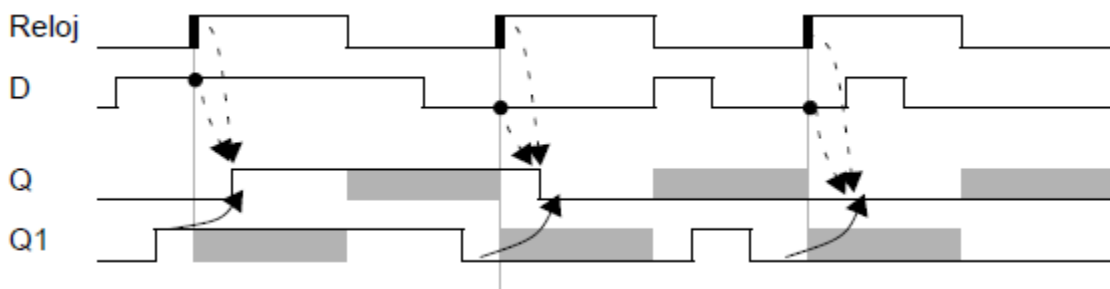
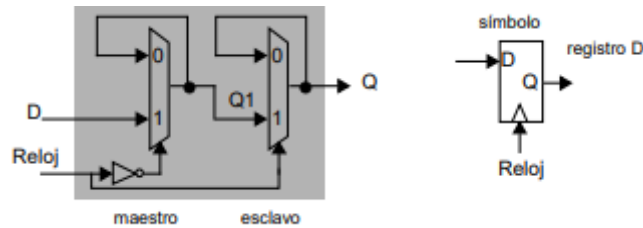
Celda D (latch D)

Las celdas trabajan por nivel de la señal de reloj. Se implementa con multiplexor, donde la salida alimenta una de las entradas, haciendo que si el reloj está en el nivel de la entrada realimentada, no se lee la entrada.



Registro D (Flip-flop D)

Los registros funcionan por flanco, se utilizan dos celdas unidas, haciendo que en el flanco que no se lee, la primera celda 'guarde' la señal y en el flanco se transmite a la segunda celda que no se actualizara hasta el siguiente flanco.



Tipos de U.F. Segmentadas

La segmentación de cada U.F. puede responder a varios patrones:

	ciclos		
etapas	1	2	3
A	X		
B		X	
C			X

a) LINEAL

	ciclos					
etapas	1	2	3	4	5	6
A	X					
B		X	X			
C				X	X	X

b) MULTICICLO

	ciclos				
etapas	1	2	3	4	5
A		X		X	
B			X		X
C	X			X	

c) NO LINEAL

En el caso de las U.F. multiciclo y no lineal, podemos tener riesgos estructurales en la interpretación sucesiva de instrucciones.

Interpretación de instrucciones Serie

Trabajaremos sobre este simple lenguaje maquina:

		Campos de la instrucción																		Descripción						
		31	...	26	25	...	21	20	...	16	15	...	12	11	...	5	4	...	0							
Clases	Tipo	6				5				5				4				7				5				
ENT	RR	CoOp				ra				rb				0	...	0	func				rc					
	RI	CoOp				ra				literal				1	func				rc							
MEM	Lo St	CoOp				ra				rb				literal												
IS	BR	CoOp				ra				literal																
	JMP	CoOp				ra				rb				func				0	...	0	0	...	0			

El camino de datos constara de un banco de registros, una memoria de instrucciones y datos, ALU, CP y un modulo de control/decodificación. En esta primera aproximación, tenemos que hay recursos que se utilizan mas de una vez en un mismo ciclo (p.e. BR). En el entorno serie, no se empieza a ejectura una instrucción hasta que se ha terminado con la interpretación anterior.

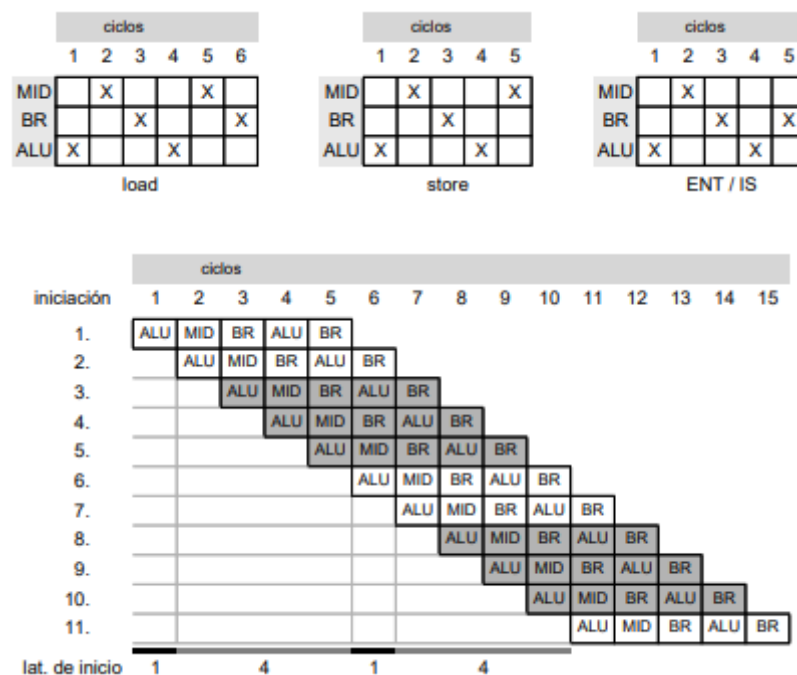
Segmentación del proceso de interpretación de instrucciones

Para reducir el tiempo de ejecución de un programa se utiliza la técnica de segmentación en la implementación del camino de datos y se solapa la interpretación de instrucciones. Hay que tener presente:

- Recursos hardware disponibles
- Respetar la semántica del lenguaje

Para diseñar un sistema segmentado primeramente hay que identificar los recursos que interviene y seguidamente las etapas en las que lo podemos dividir.

Una primera aproximación en la segmentación nos lleva a tener tablas de reserva no lineales, que pueden llevar a conflicto (riesgo estructural). La latencia de inicio son aquellos ciclos que hay que esperar entre dos tareas para que no se produzcan riesgos, en el diagrama podemos ver los ciclos que se perderian:



La latencia media de inicio en este caso seria $LMI = (1 + 4)/2 = 2.5$. Podemos ver que el minimo de la LMI sera el numero de 'X' en una fila de la tabla de reservas. Los ciclos perdidos entonces se pueden calcular como:

$$G = \frac{\text{num.Etapas} * T_c}{T_c + (LMI - 1) * T_c};$$

Ciclos perdidos: $LMI - 1$

Conflicto instrucciones con diferente | mismo patrón

Si nos interesa eliminar riesgos estructurales en un diseño, deberemos

- añadir ciclos de retardo
- añadir recursos hardware

El objetivo es que las tablas de reserva sean lineales e iguales para todas las instrucciones.

- Si un recurso se utiliza en mas de un ciclo, entonces habra que intentar replicar el recurso.
- Para que todas las instrucciones tengan las mismas tablas de reservas, habra que añadir etapas de retardo para semejar las tablas.

Para el diseño anterior, una solución sera:

	ciclos					
	1	2	3	4	5	6
+	X					
MI		X				
BRL			X			
ALU				X		
MD					X	
BRE						X

load

	ciclos				
	1	2	3	4	5
+	X				
MI		X			
BRL			X		
ALU				X	
MD					X
BRE					

store

	ciclos					
	1	2	3	4	5	6
+	X					
MI		X				
BRL			X			
ALU				X		
MD						
BRE						X

ENT

En este diseño, se ha añadido un sumador (+) para el primer ciclo, dividido la memoria en MD i MI y el banco de registros en BRL y BRE. Además se ha añadido una etapa de retardo en las operaciones de tipo ENT.

Control de un dispositivo segmentado

Deberemos determinar cuando puede iniciar una nueva tarea y controlar la lógica de cada etapa:

Control de iniciaciones

Sobre una tabla de reservas podemos observar las latencias de inicio prohibidas(latencia que da lugar a un riesgo estructural).

Si se detecta un possible fallo estructural, se debe detener el inicio o detener la interpretación.

Control lógico

Siempre debemos añadir señales de validación de los datos, pues al añadir etapas de retardo debemos evitar que se modifique el estado del procesador.

Para generar las señales de control, podemos optar por dos formatos:

- Control estacionario en el tiempo
Se trata de un elemento centralizado que en cada ciclo envía las señales necesarias.
- Control estacionario en los datos
La información de control fluye con los datos y el control esta integrado en la segmentación.
Al decodificar la instrucción se generan las señales de control neceasrias de todas las etapas.

