

Segundo Parcial Arquitectura de Computadoras 2012

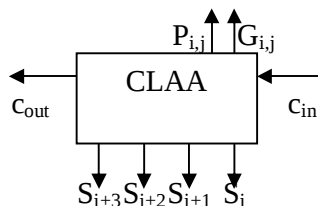
1- Esquematizar el sumador paralelo de 32 bits correspondiente a los incisos (a), (b), (c). Expresar en el diagrama los **tiempos en cada salida** de los circuitos. Calcular en cada caso el **tiempo de suma** utilizando los retardos especificados mas abajo.

- (a) En configuración *Ripple*.
- (b) En configuración *Carry Look-Ahead* de 2 niveles.
- (c) En configuración *Carry Look-Ahead* de 3 niveles.

En todos los casos emplear sumadores CLAA de 4 bits cada uno. Para los últimos dos casos utilizar los CLAG necesarios para la configuración deseada. Tener en cuenta los retardos indicados a continuación:

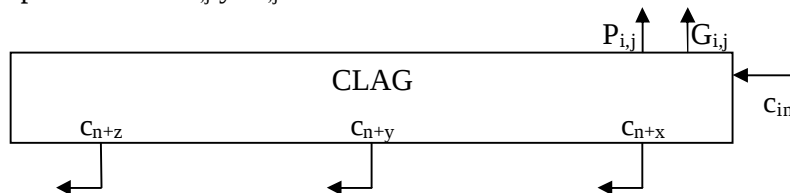
Para los CLAA:

8t para obtener la suma
6t para obtener el c_{out}
2t para obtener $P_{i,j}$ y $G_{i,j}$



Para los CLAG:

1t para obtener los c_{n+x} , c_{n+y} y c_{n+z}
2t para obtener $P_{i,j}$ y $G_{i,j}$



(d) Asumiendo una configuración de 2 niveles como la del inciso (b), desarrollar las expresiones lógicas genéricas correspondientes a la propagación ($P_{i,i+3}$) y generación ($G_{i,i+3}$) de un CLAA en función de los p_i y g_i internos y desarrollar la expresión lógica genérica del c_{n+y} de un CLAG. Finalmente a partir de las formulas anteriores, determinar el valor del carry c_{n+y} del primer CLAG para los siguientes operandos.

X=1000 0101 0000 1100 1100 0101 0101 1101

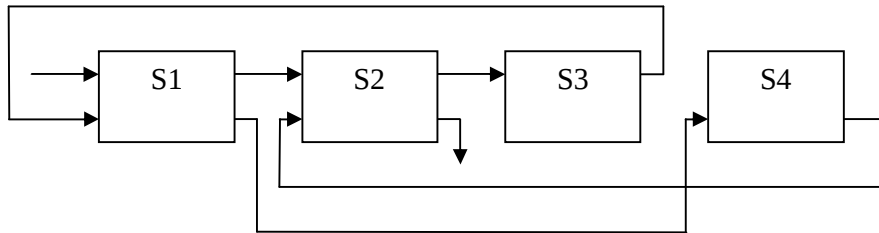
Y=0111 0011 1011 1010 0011 0110 1010 0011

2- Dados los operandos X (dividendo) e Y (divisor), trabajando con operandos de $n=6$ bits, calcular el cociente $Q=X/Y$, y el resto R de acuerdo a los siguientes ítems.

- (a) Para $X=47$ e $Y=17$, utilizando el algoritmo de división **sin restoring**. Recordar que el acumulador y la ALU deben ser de tamaño $n+1$ en tanto que el MQ será de tamaño n .
- (b) Para $X=47$ e $Y=3$, utilizando el algoritmo de división rápida **SRT** base 2. Recordar que dado que el dígito de signo es redundante, el acumulador y la ALU serán del tamaño de los operandos y se deberán analizar 2 bits y no 3.

OBS: Explicar claramente cada paso.

3- Un cierto pipeline esta organizado como se ilustra en la siguiente figura.



- Determinar la tabla de reservación asociada a la figura, las latencias críticas y el vector de colisión asociado.
- Listar **todos** los posibles ciclos de secuenciamiento con sus respectivas latencias e indicar cuales correspondiente a ciclos greedy. Identifique el/los ciclos greedy con latencia promedio minima. ¿Podría llegar a mejorarse el secuenciamiento encontrado? Justificar.
- Ensayar la incorporación al pipe de una etapa de retardo S_5 de forma tal que el dato saliente de la etapa S_4 ingrese a la etapa de retardo S_5 en el sexto ciclo y luego continúe con el secuenciamiento original. Compare los resultados obtenidos para el pipe original, tanto a nivel de throuput como a nivel de tiempo de ejecución de una instrucción individual.
- Analizar la evolución de los primeros 6 datos en el nuevo pipe al ser secuenciados según el régimen optimo. ¿Cómo visualiza en este diagrama la posibilidad o no de mejoras?

4- Un procesador cuenta con un pipe de instrucciones de 5 etapas como el considerado en la práctica. Además dispone de una unidad aritmética de multiplicación de 4 ciclos y una unidad para la suma de 1 ciclo. Asumiendo que durante la primera mitad del ciclo se escribe en el banco de registros y durante la segunda mitad se lee y que los conflictos de nombre se resuelven por medio de renombramiento de registros, desarrollar el **diagrama de Gantt** correspondiente a la evolución de la siguiente secuencia de instrucciones y calcular el **CPI promedio alcanzable** en cada uno de los escenarios dados a continuación. Explique porque se producen e indique claramente los ciclos Stall y Wait que surjan. Marque claramente los forwardings requeridos según corresponda.

I_1	MULT R_4, R_2, R_1
I_2	ADD R_5, R_4, R_3
I_3	MULT R_9, R_5, R_6
I_4	LOAD $R_7, 0(R_{15})$
I_5	ADD R_{10}, R_9, R_7
I_6	ADD $R_1, R_{11}, \#8$

- Ejecución en orden, sin forwarding.
- Ejecución en orden, con forwarding.
- Ejecución fuera de orden, con forwarding. Asumiendo que cada unidad funcional cuenta con estaciones de reservación asociadas para implementar la ejecución fuera de orden, determine cuantas estaciones de reservación serán necesarias en el multiplicador y en el sumador de forma de evitar los ciclos Stall.