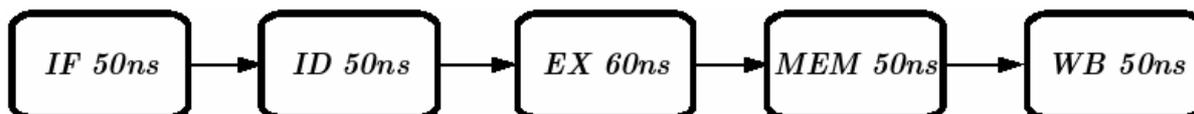




## Consideraciones sobre la profundidad del pipeline de instrucciones

Veamos como podría resultar el pipeline del procesador **RISC DLX** si asignamos un segmento, etapa del pipe, a cada una de las cinco fases. Asumamos, a título de ejemplo, que demandan: **FETCH**  $50ns$ , **DECODE**  $50ns$ , **EXECUTE**  $60ns$ , **MEMORIA**  $50ns$ , y **WRITE BACK**  $50ns$ . A su vez, que los latches interetapas agregan  $5ns$  de retardo adicional. El pipe descripto guarda la siguiente estructura:



Dado que trabajamos de forma sincrónica a la velocidad de la etapa más lenta, se tiene un período de  $65ns$ . Considerando que sin pipe el tiempo por instrucción es de  $260ns$ , resultará un speedup de 4, contra el 5 teórico (del número de etapas). Ha incidido para ello la disparidad entre etapas y el retardo del latch en cada una. Es un hecho de que si avanzamos en el sentido de incrementar el número de etapas en un pipeline se transita un camino que posibilita reducir el período de reloj, esto es menor número de niveles de compuertas en el camino de la señal en un ciclo, con lo cual uno de los ejes de la performance, en este caso la frecuencia de reloj, se verá favorecida. Esto lo podemos ver en el **Pentium IV** al momento que Intel, visto la limitación para superar 1GHz con el **Pentium III** (por caso, recordemos el fiasco con los Pentium III de 1.12Ghz), concibió esta nueva arquitectura, donde se "abusaba" del número de etapas.

La cuestión, y por eso el concepto "abusaba", es que se debe alcanzar un equilibrio en la profundidad del pipe, dado que otro pilar de la performance son los ciclos efectivos por instrucción **CPI**, el cual como se verá se resiente, aumenta, con pipe más profundos.

Pero más allá de esta consideración que es perfectamente válida, la pregunta que surge es ¿Cuán profundo se puede hacer un pipeline? ¿Es una cuestión sin límite, más allá de las ventajas que se tengan, o no, al hacerlo?

La respuesta es ¡**NO!**

Si bien en una primera apreciación se podría pensar que no existe límite alguno, veremos que en realidad esto no es así. Un requerimiento de mínima que se debe satisfacer, el cual se podría sumar a los requerimientos básicos vistos al principio para un pipe de instrucciones, es el de poder ejecutar las operaciones elementales back to back, consecutivamente, esto es que el resultado en el procesamiento de una instrucción pueda ser utilizado por la inmediata siguiente, asegurando la fluidez de instrucciones en el pipe.

Esto fija un límite claro al menor número de niveles a atravesar en un ciclo de reloj, el cual sería el de la implementación óptima de un circuito sumador, alrededor de 7 niveles de compuertas para un hardware de 64 bits. Se requiere una operación atómica de esta unidad, esto es, que se resuelva en un ciclo.

Además de este límite inferior se deberán analizar aquellas etapas que realizan otras funciones que deban ser atómicas, como se vio en el sentido de no frenar la ejecución consecutiva de instrucciones. En la medida que éstas demanden más tiempo que la operación de suma se constituyen en un cuello de botella para el pipe, vale decir fijan un tiempo de ciclo mayor al mínimo correspondiente a la operación básica de suma.