

Supertomasim es un simulador visual desarrollado como un proyecto Fin de carrera dentro del Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Sevilla: SUPERTOMASIM (SUPERescalar TOMasulo SIMulator). Éste simula un procesador didáctico superescalar y superencadenado del DLX, que usa el algoritmo de Tomasulo de “scheduling” dinámico, siendo configurables tanto el grado de superescalaridad como los recursos del procesador (UF, CDB, RS, etc).

Algunos detalles de la interfaz de usuario están aún incompletas, son poco amigables o no detectan ciertos errores de entrada. Por tanto, cuando escriba código DLX, debe tener en cuenta los siguientes pormenores (se irán depurando en las siguientes versiones del simulador):

- **No puede haber una línea en blanco al final del fichero.** No distingue entre mayúsculas y minúsculas.
- Detrás de la directiva *.text* (cuando se escriba el código), **no se pueden poner líneas en blanco**, ni líneas que comiencen por “;” (es decir, comentarios que ocupen toda la línea).
- Las **constantes de un direccionamiento** deben ir a la izquierda del paréntesis. Así: `sw -4(r1), r11`
- Las **etiquetas de salto deben empezar en el primer carácter de la línea y no pueden estar solas en una línea**, sino que detrás del “:” debe ir una instrucción válida; por ejemplo, así:
`bucle: lbu r3, 0(r1) ; Carga Arrayx[i]`
- Los registros flotantes de doble precisión se llaman `d0`, `d1`, `d2`, etc. (no son válidos los `f0`, `f2`, etc.).

Cuando ejecute las simulaciones tenga en cuenta:

- Para **cambiar cualquier parámetro de la configuración** (cuadro de diálogo de configuración: *ventanas->configuración*) o **para cambiar de código simulado**, debe primero resetear el simulador con la opción: *Archivo-> Limpiar*. Se pierde evidentemente todo el cronograma anterior.
- No use la **ejecución completa: es muy lenta**. Ejecute **paso a paso o de 5 en 5 pasos** (teclas F7, F8). Además, en la ejecución completa las estadísticas incluyen las instrucciones finales (NOP y trap) y por tanto, las estadísticas que nos daría el simulador no serían las buscadas.
- Para **cargar una predicción** para los saltos, escribir un fichero de nombre *btb.txt* que contenga en cada línea una dirección en hexadecimal (comenzando por 0X), el símbolo “;” y las siglas N o NT. **Tampoco se pueden poner líneas en blanco**, ni líneas que contengan otros símbolos, ni puede haber una línea en blanco al final del fichero. Por ejemplo, un fichero *btb.txt* puede ser:
`0X2C;NT`
`0X34;NT`
`0X3C;NT`
`0X44;T`

Al ejecutar un código en el simulador, la ventana *ventanas->BTB* informa únicamente de las predicciones que ya se han realizado (para los saltos que han ejecutado su fase IF).

- **Las UF no están segmentadas**, por tanto para emular su segmentación, aumente el número de UF hasta tantas como su duración (hacer el parámetro Unid Func igual al Num Fases EXE).
- **El número máximo de RS de cada tipo es 150. El número máximo de UF de cada tipo es 10.** Podría interrumpirse la simulación si se superan estos límites (no se chequean estos rangos en la configuración).

SuperTomasim permite configurar en el menú *Ventanas->Configuración* (cuadro de diálogo inferior derecho):

- Duración de cada UF (Num Fases EXE)
- Número de RS (Num RS)
- Número de UF (Unid Func)
- Grado de Superescalaridad (Grado Escalaridad)
- Grado de superencadenamiento de la fase IS (Num Fases IS)
- Número de CDB (Num CDB)
- Fase en la que se realiza la predicción de la BTB (Pred BTB en Fase)

Algunas consideraciones la arquitectura que simula SUPERTOMASIM son:

- En cuanto a reglas de emisión, actualmente sólo admite la **emisión Flexible**, no existiendo el concepto de tubería.
- La ventana de instrucciones es siempre **deslizante**.

- **No simula la evolución de BTB**, pero si predice saltos en función del fichero *BTB.txt* (se explica más adelante)
- **Especula** con las instrucciones sucesivas al salto, y las **aborta tras la fase WB del salto** si la predicción de un salto fue errónea (introduciendo con el símbolo “AB”), aunque no implementa buffer de reordenación. Las instrucciones especulativas que serán abortadas por error de predicción muestran su cronograma y reservan RS, pero no modifican el estado del procesador (no escriben nunca ni en la memoria ni en los registros), y por lo tanto puede que no se ejecuten correctamente (esto no afectará al resultado del código ni al rendimiento IPC, ya que estas instrucciones se abortarán de cualquier forma).
- **Cada bloqueo estructural se muestra repitiendo la fase** que no puede avanzar. Por ejemplo: IF IF IF IS significa que han habido dos ciclos de bloqueo porque IS no pudo ejecutarse.

El **cronograma y configuración se pueden salvar** en ficheros aparte (el cronograma tiene cada fase separada por tabuladores para que se pueda copiar fácilmente por ejemplo en una hoja de Excel).

Además, en el **fichero de Ayuda.pdf** se puede leer una descripción básica de su manejo.