

Modelagem de Sistemas em Silício

2017/01

Trabalho: Modelagem do RISC16

Modelar o processador RISC16, cuja especificação se encontra em PDF anexo, utilizando dois tipos de modelos:

1. **Modelo com THREADS:**

Modelar os processos fundamentais *fetch* – *decode* – *execute* na forma de SC_THREADS sincronizadas por eventos. Utilizar como referência o código C do RISC16 enviado em anexo no mural da disciplina.

Incluir uma *thread* auxiliar *start* para sincronização das demais *threads*, de forma que a primeira *thread* a entrar em execução seja a *fetch*, que busca uma instrução na memória. Depois de escrever a instrução lida da memória em RI e incrementar o PC, a *fetch* notifica a *thread decode*. Esta extrai os campos da instrução e sinaliza a *thread execute* para que processe a instrução.

2. **Modelo com Módulos interligados por filas Bloqueantes**

Criar um modelo onde as fases de uma instrução são implementadas por módulos. Neste caso, *fetch*, *decode* e *execute* são módulos que implementam as fases de execução de uma instrução sincronizados por filas bloqueantes. As filas devem transmitir um ponteiro para o contexto de execução de uma instrução. O contexto inclui as informações necessárias à execução da instrução.

Por exemplo, o contexto de uma instrução deve incluir ao menos as seguintes informações:

- pc (contador de programa), ri (registrador de instruções), opcode, rs, rt, rd, k4, k8

A memória e o banco de registradores devem ser implementados como módulos à parte. O módulo *fetch* precisa ler a memória e o módulo *execute* precisa ler e escrever na memória. Uma forma simples de implementar essa conexão é criando uma interface de comunicação a ser utilizada em uma porta de entrada e saída nos módulos *fetch* e *execute*. A memória implementa a interface, como se fosse um canal de comunicação, e pode ser diretamente conectada à porte dos módulos *fetch* e *execute*. A mesma solução pode ser utilizada para conectar o banco de registradores aos módulos *decode* e *execute*.

Verificação: utilizar a função que gera instruções para testar a execução correta de cada instrução do processador. Isso deve ser testado em ambos os modelos.

Entrega:

- Descrição textual da implementação
- Descrição da verificação do funcionamento
- códigos SystemC