

Entrega 4 - Relatório

Grupo - Popullacho (Placa 4)

Alexandre Ladeira	212328
Gabriel Pallotta Cardenete	216392
Igor Fernando Mandello	236769
Júlia Alves De Arruda	238077
Lucas Hideki Carvalho Dinnouti	220792
Vinícius Waki Teles	257390

Organização

Nas últimas etapas havíamos nos divididos em 3 grupos para facilitar e acelerar o desenvolvimento, sendo o grupo A composto por Alexandre e Gabriel, grupo B por Igor e Vinícius, grupo C por Júlia e Lucas. Nessa etapa, os grupos A e B trabalharam mais próximos para finalizar as últimas implementações e o Grupo C para testes finais dos periféricos.

Grupo A e B

Responsabilidades:

- Implementação da decodificação de instruções compactas
- Implementação de ferramentas para compilação de programas alto nível (C) para RISC-V, e criação de programas de teste

Grupo C

Responsabilidades:

- Tradução do programa de periféricos para a linguagem C
- Compilação do programa com instruções compactas e testes na FPGA

Desenvolvimento do projeto

Instruções compactas

Para essa entrega, implementamos a decodificação para todas as instruções compactas, complementando o que já havia sido desenvolvido na última entrega.

Adicionamos um trecho de código no módulo “decoder” que realiza a decodificação se a instrução lida é compacta, a partir dos primeiros 2 bits da instrução. Essa decodificação considera apenas os 16 primeiros bits da instrução, e como toda instrução compacta pode ser mapeada diretamente para uma instrução normal, não foi necessária nenhuma mudança nas saídas do “decoder”.

Uma mudança adicional necessária foi para o suporte de instruções JAL compactas, que escrevem o endereço da instrução seguinte no registrador. Para instruções compactas, esse endereço é o endereço atual somado de 2, ao invés de 4 para instruções tradicionais.

Periféricos

Na última entrega ficou decidido que a aplicação dos periféricos seria um radar de velocidade, no qual os sensores de luz são usados para detectar a passagem de um objeto e os LEDs para indicar se o tempo da passagem entre os dois sensores foi alto ou baixo. Sendo vermelho quando a velocidade passa da permitida e verde quando está dentro do permitido.

O programa para esse radar tinha sido primeiramente construído em assembly/risc-v. Nessa entrega o programa foi reescrito em linguagem C e compilado para risc-v antes de ser gravado na placa. Com isso, temos uma aplicação completa para os periféricos, do programa ao hardware.

Ferramentas de compilação

Também implementamos um script que realiza a compilação de um programa em alto-nível na linguagem C, para o conjunto de instruções RV32IMAC, no formato hexadecimal lido do Verilog. Esse script utiliza o ferramental RISC-V gnu, que deve ser compilado na máquina local, ou utilizando uma imagem Dockerfile provida no repositório.

Com esse script, pudemos implementar alguns programas de teste que utilizam diversas instruções RISC-V, inclusive instruções compactas, e incluímos esses programas na suíte de testes automatizados desenvolvida para entregas anteriores.

Lições aprendidas

Tivemos dificuldades na implementação e compilação do programa em C para risc-v. Principalmente quanto à parte de memória, por vezes escrevemos versões ineficientes do programa que acabava com o espaço que tínhamos disponível.