**MÓDULOS EXISTENTES:**

**i\_cache.v:** código gerado para cyclone IV E, de uma memória do tipo ROM, com 2048 palavras de 32 bits. O arquivo de inicialização tem por nome rom.mif e deve estar na mesma pasta em que o arquivo de projeto do quartus estiver. Esta memória tem a função lógica de memória de programa para o RISCV;

**d\_cache.v:** código gerado para cyclone IV E, de uma memória do tipo RAM, com 4096 palavras de 32 bits. O arquivo de inicialização tem por nome ram.mif e deve estar na mesma pasta em que o arquivo de projeto do quartus estiver. Esta memória tem a função lógica de memória de dados para o RISCV;

**riscv\_regfile.sv:** implementação do banco de registradores. O banco sempre recebe dois endereços e libera dois valores de registro, e quando usado para escrita, recebe um endereço de destino e o valor de registro a ser escrito e um sinal de write. Complementarmente, libera os valores de return address (ra), stack pointer (sp), e result (result, o resultado de operações guardado no registrador temporário mais usado pelo compilador);

**riscv\_alu.sv:** implementação da unidade lógica-aritmética, tendo como submódulos: divisor (**riscv\_div.sv**) e comparador (**comparator.sv**). A ULA realiza operações de soma, subtração, multiplicação, divisão, comparação, shift (left lógico, right lógico e aritmético), or, and e xor. Além das entradas esperadas - operadores e seletor de operação - tem-se a flag para habilitar o módulo da divisão e como saídas, uma flag para avisar quando a operação de divisão encerrar (freeze\_pipe). Não há flags para indicar overflow ou underflow.

**inst\_mem\_ctrl.sv**: interface entre a memória de programa (i\_cache.v) e o módulo top da CPU, para realizar o ajuste de endereço e, se preciso, de dado.

**mem\_ctrl.sv**: interface entre a memória de dados (d\_cache.v) e os módulos do barramento e da CPU (módulo top), para realizar o ajuste de endereço e de dado.