

Laboratório 04 - Grupo 09

- CPU femtoRISC-V MULTICICLO -

Antônio Vinicius de Moura Rodrigues 19/0084502,
Gabriel Pinheiro da Conceição 19/0133724,
Leandro de Sousa Monteiro 17/0060454

¹ Universidade de Brasília - Instituto de Ciências Exatas
Dep. Ciência da Computação - CIC0099 - Organização e Arquitetura de Computadores
2020.2 - Turma A - Professor Marcus Vinicius Lamar
Prédio CIC/EST - Campus Universitário Darcy Ribeiro
Asa Norte 70919-970 Brasília, DF

1. ULA

A ULA utilizada neste relatório necessitou processar operações de até 32 bits. O Deeds oferece uma opção de ULA já implementada de 16 bits, desse modo precisou-se fazer algumas adaptações de maneira que ela pudesse se adequar ao projeto.

Foram então utilizadas duas ULAs de 16 bits e as modificações adicionais podem ser observadas abaixo:

1. As entradas que definem a operação a ser realizada pelo componente foram compartilhadas entre as duas.
2. As entradas dos operandos A e B responsáveis por fornecer à ULA os argumentos para cálculo foram divididos em dois BUSES de 16 bits cada e distribuídos entre as duas entradas para cada argumento de cada ULA.
3. A saída CO da primeira ULA, ou seja, a ULA que opera com os primeiros 15 bits de cada argumento foi conectada na entrada CI da segunda ULA de modo que as operações feitas possam ter continuidade com base no *carry out* da primeira ULA.
4. A saída das duas ULAs foi concatenada de modo a compor o resultado final da operação com 32 bits

O resultado final da ULA pode ser visto na figura 1 que representa o circuito interno do CBE utilizado no processador *femtoRISC-V* Multiciclo.

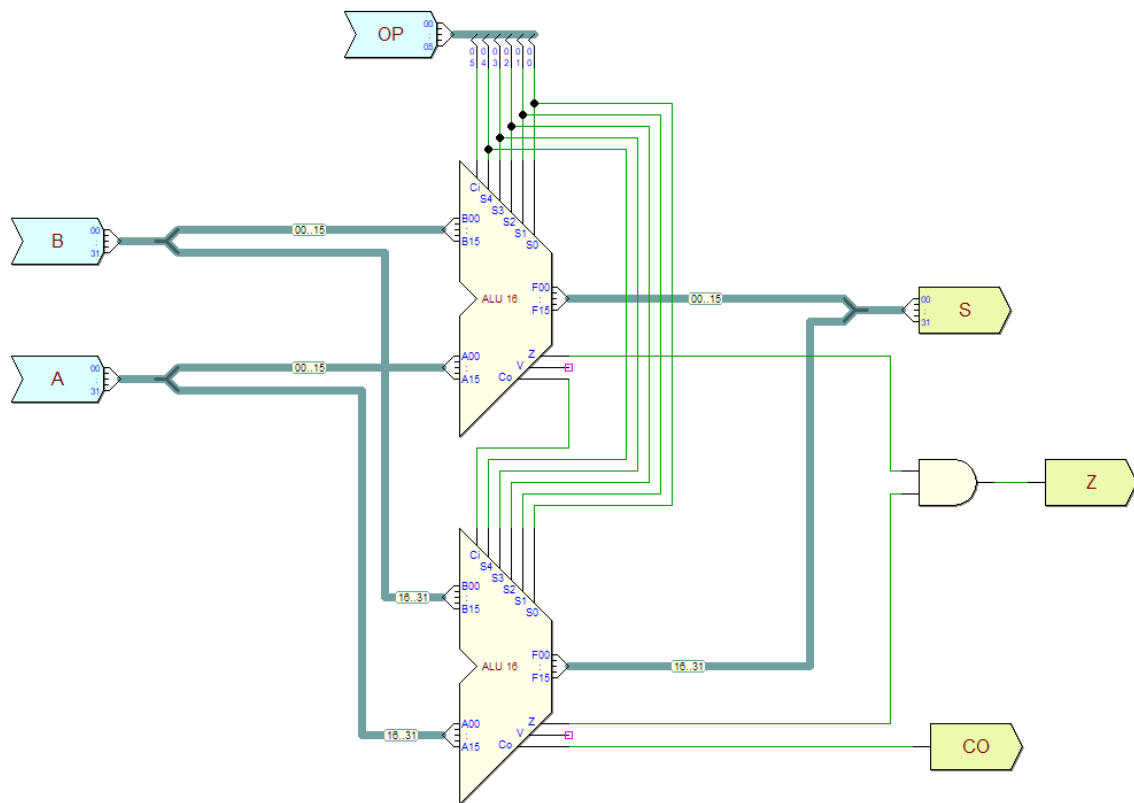


Figure 1. Circuito interno do CBE da ULA

2. Shift Left Logical Immediate

Para a realização do Shift Left Logical Immediate (SLLI) foi preciso pegar a entrada de 32 bits que vem do registrador "A" e deslocá-lo para a esquerda a quantidade de vezes do imediato que vem no registrador "B". Para isso é necessário pegar o bus da entrada "A" e manipular manualmente os bits para que eles fiquem deslocados, isso é realizado para todas as 32 opções de deslocamento possíveis. Depois disso basta utilizar multiplexadores que recebem os 5 primeiros bits do registrador "B" para escolher qual das 32 opções de deslocamento que devem estar na saída

O resultado final do circuito interno do CBE utilizado pode ser visto na figura 2.

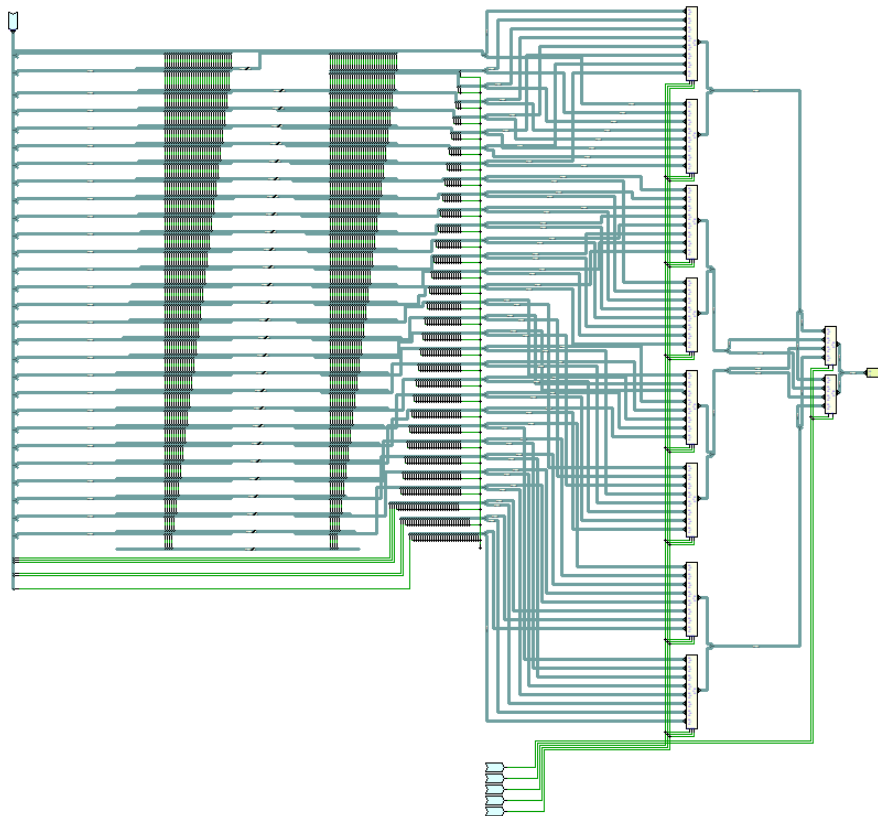


Figure 2. Memória de Dados

3. Banco de Registradores

Para a construção do banco de registradores foi usada como base a figura referência de [Pattersson and Henessy 2021] que idealiza um banco de registradores com 32 registradores de 32 bits cada. Além disso, o banco possui duas entradas de endereçamento utilizadas para seleção dos registradores, uma entrada de endereçamento para seleção do registrador de destino das operações realizadas com outros registradores, uma entrada de habilitação da escrita nos registradores e uma entrada ligada diretamente à ULA e à RAM que permite a transferência de argumentos recentemente calculados ou armazenados na RAM diretamente ao banco. A idealização de [Pattersson and Henessy 2021] inclui ainda duas saídas de dados, o que permite a leitura de dois registradores de forma simultânea. Um desenho esquemático do banco de registradores implementado pode ser visto na figura 3 também retirada de [Pattersson and Henessy 2021].

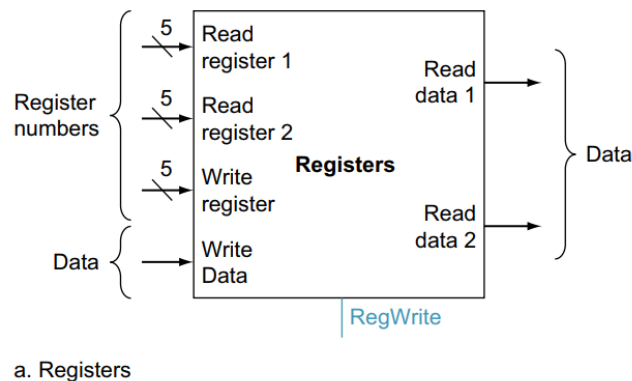


Figure 3. Desenho esquemático do banco de registradores

Para a seleção dos registradores a seguinte lógica foi aplicada: Foram utilizados 8 sets de 2 mux ($8 \rightarrow 1$) cada e 6 sets de 2 mux ($2 \rightarrow 1$) cada. Todos os mux utilizados possuem 16 bits e por isso a necessidade de se usar 2 mux para cada seleção desejada. Os mux foram dispostos de modo a fazer uma filtragem por setor de registradores.

Os setores foram divididos da seguinte forma:

1. Dois setores principais divididos entre esquerda e direita, cada um com 16 registradores.
2. Dentre de cada setor principal os registradores foram divididos em mais dois setores: Superior e Inferior, cada um com 8 registradores.
3. Dentro dos setores superior e inferior cada registrador pode ser selecionado por um número de 0 a 7.

Os 3 bits de seleção menos significativos são os responsáveis por filtrar os registradores do número correspondente em cada um dos 4 setores superior e inferior dos lados esquerdo e direito do banco de registradores. O próximo bit de seleção é o responsável por escolher entre o setor superior ou o inferior de cada lado e o último bit (MSB) seleciona o lado esquerdo ou o direito.

O número 00001, por exemplo, seleciona o registrador 1 da seguinte forma:

1. 0 seleciona o setor esquerdo ou o direito. Nesse caso será o setor esquerdo.
2. 0 seleciona o setor superior ou o inferior de cada lado. Nesse caso será o setor superior.
3. 001 = 3 bits menos significativos, selecionam o registrador 1 em cada um dos 4 setores superior e inferior.

Com essa lógica foi possível então implementar o banco de registradores no Deeds. O resultado final pode ser visto na figura 4. Recomenda-se que para uma visualização mais detalhada do banco de registradores, o arquivo cbe correspondente seja consultado. Na imagem os setores foram separados por cor de forma a facilitar a visualização da lógica de seleção.

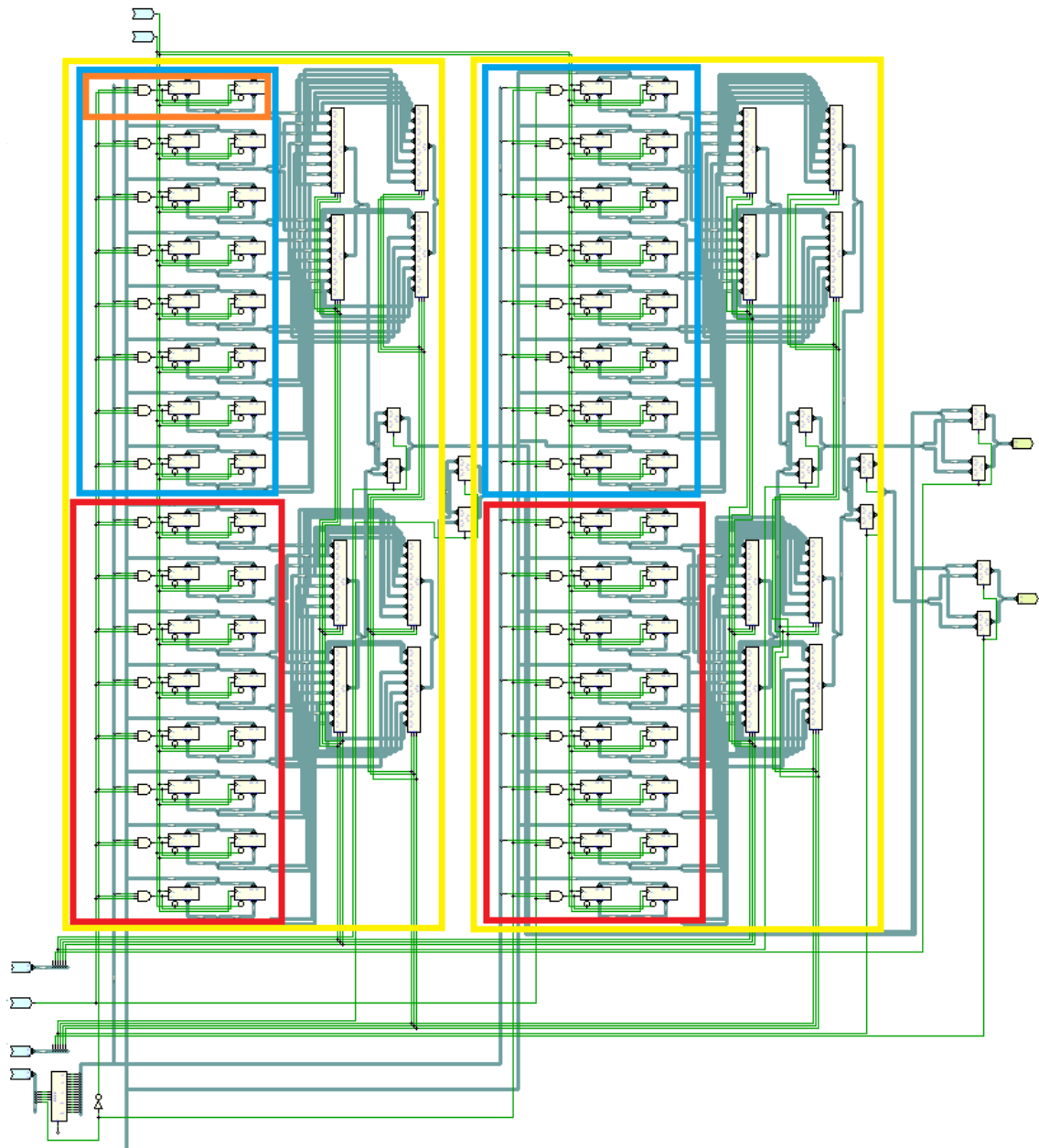


Figure 4. Banco de Registradores

- Amarelo: setores esquerdo e direito
- Azul: setores superiores
- Vermelho: setores inferiores
- Laranja: registrador individual de cada setor selecionado por 3 bits

4. Datapath e bloco de controle

O datapath utilizado para a construção do processador *femto*RISC-V Multiciclo foi retirado de [Patterson and Hennessy 2021] e seguido à risca de modo que se assemelhasse de maneira mais fiel possível à versão do autor. A figura 5 apresenta um desenho esquemático do caminho de dados padrão sem modificações utilizado.

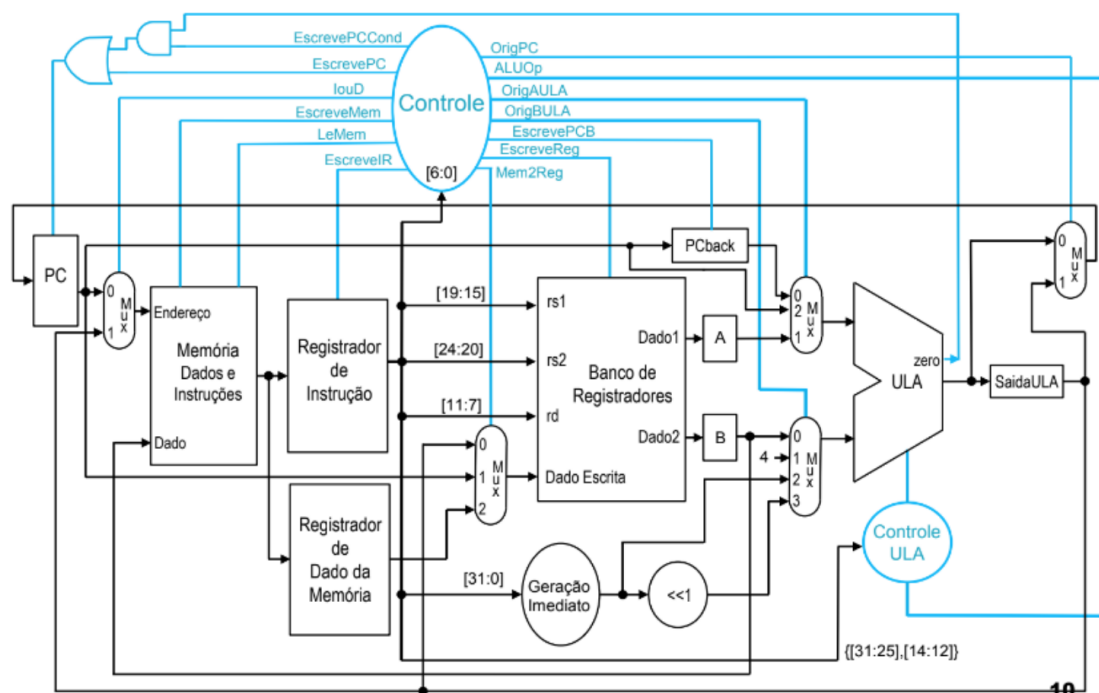


Figure 5. Caminho de Dados

4.1. Gerador de Imediato

Na geração do imediato de cada instrução foi utilizado um circuito decodificador que atribui um número a cada opcode da instrução submetida ao circuito. O número foi então utilizado para fazer a seleção de dois mux ($8 \rightarrow 1$) que contem em cada entrada o imediato gerado para cada instrução. Os immediatos foram gerados com base na tabela da figura 6 retirada de [Pattersson and Henessy 2021] manualmente, ou seja, reorganizando bit a bit e completando o restante dos bits com o valor 0 para que o imediato possua os devidos 32 bits necessários à configuração do processador.

	31	27	26	25	24	20	19	15	14	12	11	7	6	0
R	funct7				rs2		rs1		funct3		rd		opcode	
I	imm[11:0]					rs1		funct3		rd		opcode		
S	imm[11:5]				rs2		rs1		funct3		imm[4:0]		opcode	
SB	imm[12 10:5]				rs2		rs1		funct3		imm[4:1 11]		opcode	
U	imm[31:12]										rd		opcode	
UJ	imm[20 10:1 11 19:12]										rd		opcode	

Figure 6. Formatos das instruções da ISA RISC-V

A figura 7 apresenta o resultado final do circuito interno do CBE implementado para a geração de imediatos. Recomenda-se o exame do arquivo CBE original para a observação de mais detalhes quanto ao circuito em questão.

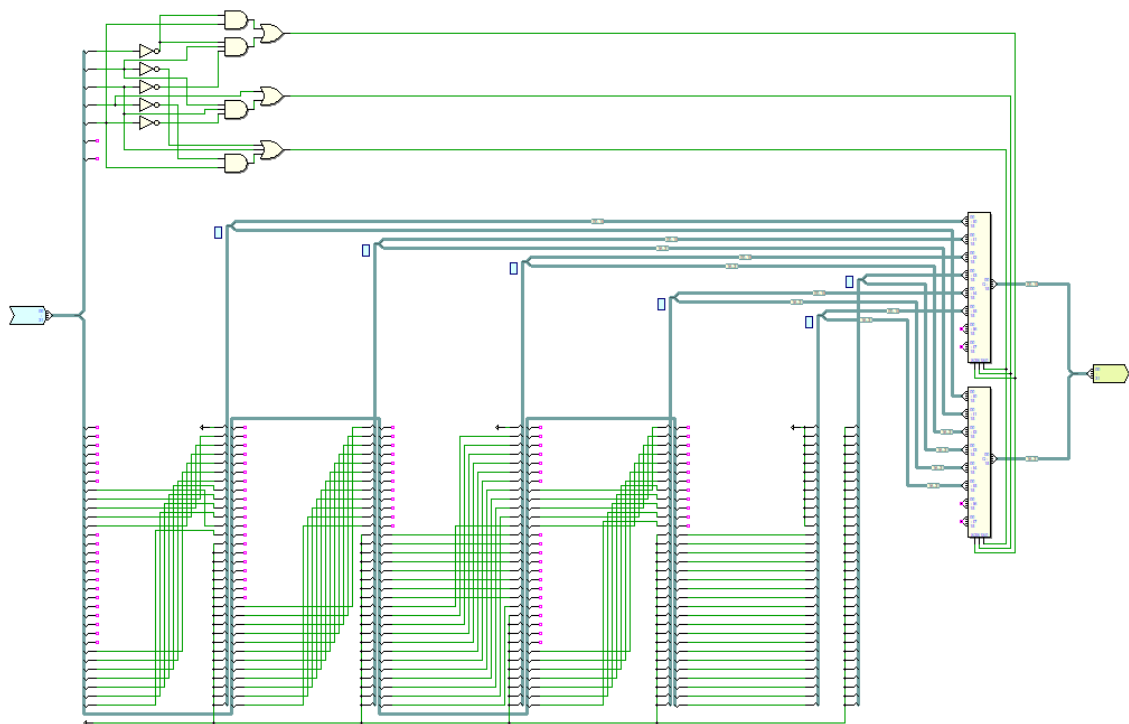


Figure 7. Circuito interno do CBE gerador de imediato

4.2. Multiplexador de 32 bits

Na implementação do processador em questão foi necessária a utilização de diversos multiplexadores ($2 \rightarrow 1$) de 32 bits cada. No entanto, o Deeds não fornece um mux que seja capaz de multiplexar tantos bits e desse modo foi necessário implementar um CBE que realizasse a tarefa.

O circuito foi feito utilizando a divisão e concatenação de bits dos BUSES de entrada e saída. A figura 8 apresenta o resultado final do circuito interno do CBE implementado.

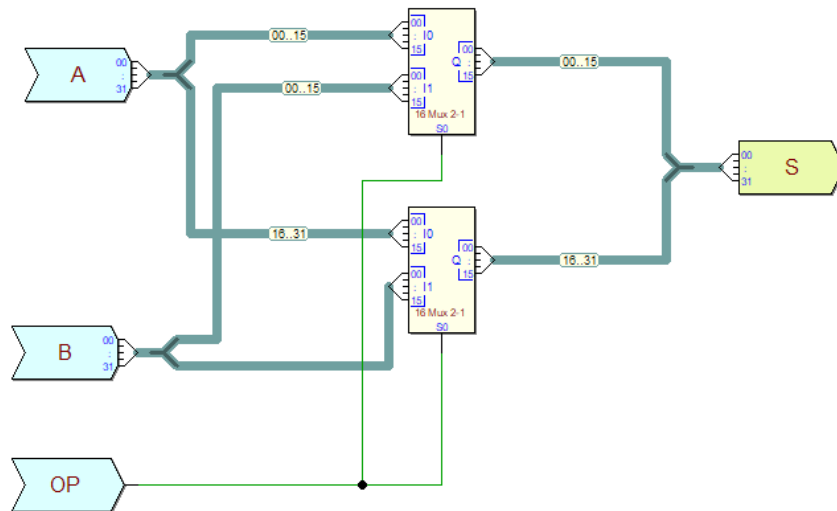


Figure 8. Circuito interno do CBE do Mux de 32 bits

5. femtoRisc-V Multiciclo completo

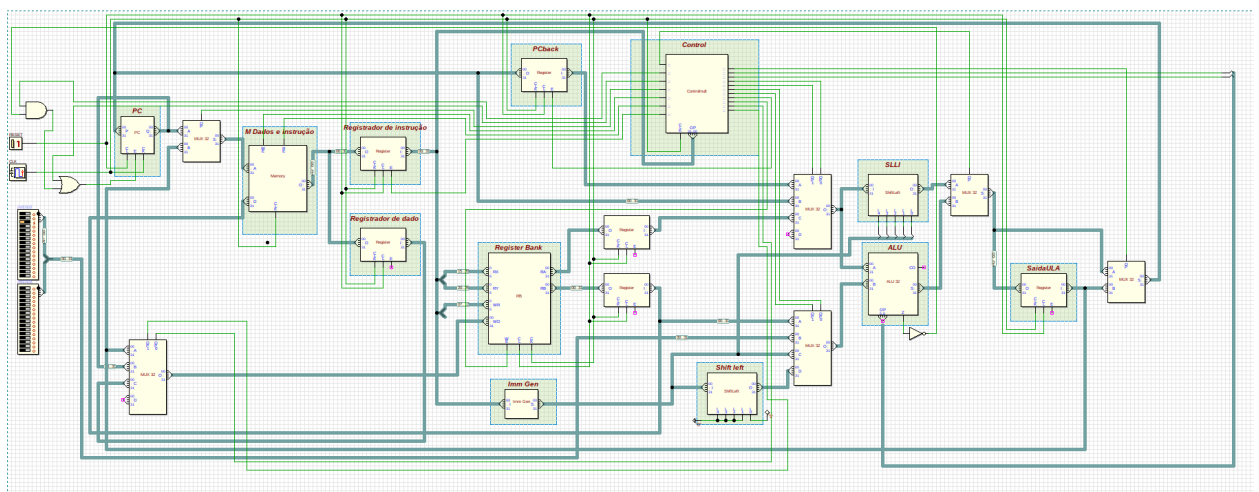


Figure 9. Circuito do *femtoRisc-V* Multiciclo

O processador completo não foi capaz de executar todas as instruções, pois nem todos os blocos funcionaram corretamente, com isso a montagem completa do processador apresentou problemas. Os Blocos que não funcionaram de maneira adequada foram a memória de dados e instruções e o controle.

References

[Patterson and Hennessy 2021] Patterson, D. A. and Hennessy, J. L. (2021). *Computer Organization and Design RISC-V Edition*. Elsevier Inc.: Katey Birtcher, 2nd edition.