

Descrição de implementação da ISA:

O projeto foi baseado no Redux V, principalmente a ordem dos opcodes, porém tomei a decisão de utilizar 8 registradores e utilizar comandos de Branch inspirados no Risk V. Além disso, há mudanças consideráveis nos formatos, sendo eles: R, B, I, J.

No formato Register, optei por 3 opções de registradores, diferente do Redux V, contudo, para isso eu sacrifiquei um bit do registrador Rs2, isto é, apenas os 4 primeiros registradores podem operar nesta posição.

Nos formatos Branch e Immediate: tomei a decisão de utilizar apenas 2 bits para o imediato, assim, para tomar saltos maiores eu tirei o complemento de 2, fazendo com que o imediato fique no intervalo [0, 3] ao invés de [-2, 1].

No formato Jump: nesse caso eu tomei a decisão de deixar o imediato o maior valor possível, 5 bits, e manter o complemento de 2. É válido ressaltar que optei por definir a operação Addi como jump, tornando o rd o registrador operando, com isso, fica indispensável o registrador Zero como uma constante na implementação, uma vez que todos os registradores antes de receberem o imediato devem ser inicializados com valor zero, utilizando a operação Add com o registrador Zero operando.

Por fim, o comando de parada deve ser feito por meio de um Branch, pois os 16 comandos possíveis foram preenchidos, assim deixando de fora um comando específico para a parada.

Listagem dos componentes utilizados:

- ULA: 1 mux de 4 bits de seleção, 2 shifters ;
- PC: 1 registrador de 12 bits ;
- Banco de Registradores: 1 demux de 3 bits de seleção, 2 mux de 3 bits de seleção, 7 mux de 1 bit de seleção ,7 registradores de 12 bits.
- ISA (circuito final): 3 somadores de 12 bits, 6 mux de 1 bit de seleção, 5 mux de 2 bits de seleção;