**Relatório do laboratório 11 - Unidade de Controle do Neander**

Nome: Arthur Ferreira Ely

Cartão: 338434

Data: 19/03/2023

Número da sala: 103

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

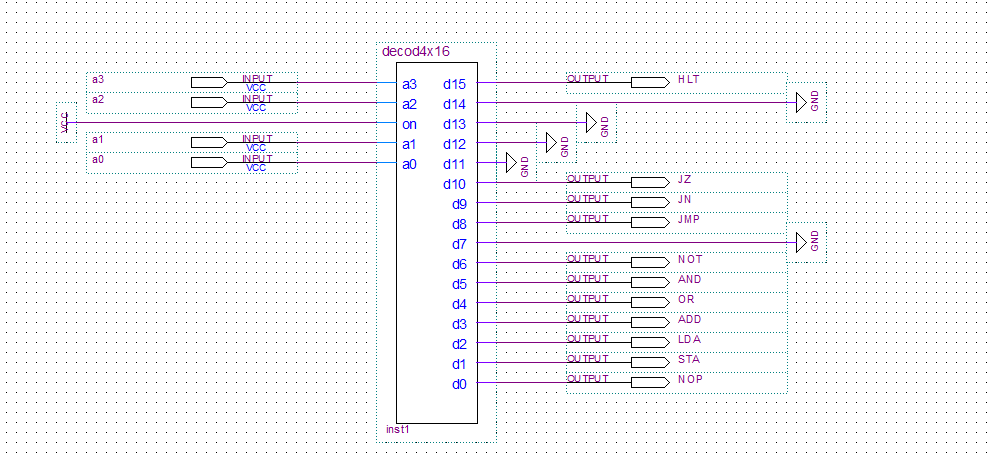
Foi realizado, neste laboratório, a implementação da parte de controle do Neander. A parte de controle serve para mandar sinais para o resto do computador, para que assim possamos deixa-lo completamente funcional. A metodologia seguida foi a do livro do professor Weber, fazendo toda a lógica utilizando um contador de três bits e dois decodificadores. Os Flip-Flops utilizados foram do tipo D.

**Células da unidade de controle (UC)**

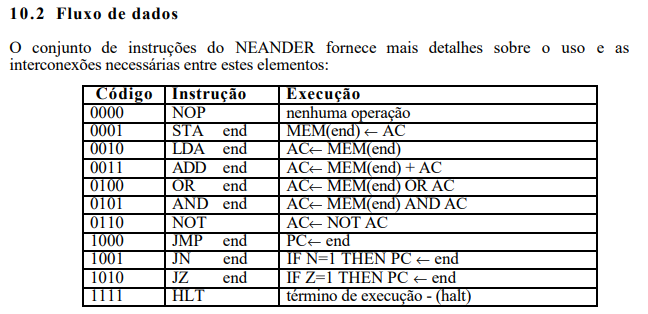
Para o desenvolvimento da UC, foi adotado a técnica *bottom-up,* fazendo pequenos componentes lógicos e os juntando na UC completa posteriormente. A parte de controle é formada por:

* Decodificador das instruções;
* Contador de três bits;
* Decodificador dos sinais de controle.

**Decodificador das instruções**

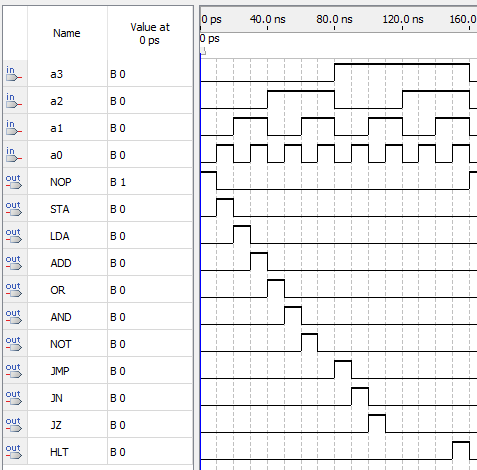
Serve para dizer qual instrução está vindo da memória conforme os 4 bits mais significativos do valor que está no registrador de instruções (RI). A seguir, a figura 1 mostra o diagrama esquemático do decodificador, baseado na figura 2, do livro do professor Weber. As entradas que não resultam em nenhumas instruções foram conectadas à *ground* (GND).

**Figura 1: decodificador das instruções do Neander.**

****

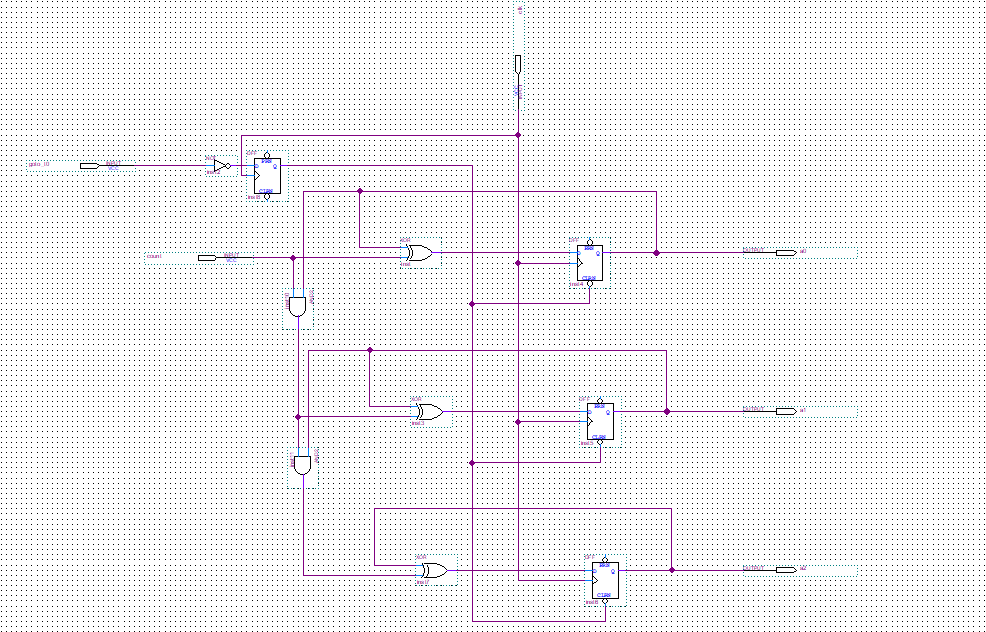
**Figura 2: tabela com os códigos das instruções do Neander.**

A seguir, a simulação desse decodificador:



**Figura 3: simulação do decodificador das instruções.**

**Contador de três bits**

Serve para temporizar toda a lógica de controle. A cada borda de subida do clock, o contador vai ser somado em 1. Cada tempo fará algo diferente, seguindo o modelo do livro do professor Weber. A seguir, a figura 1 mostra o diagrama esquemático do contador. Um adicional desse contador de três bits é o *goto\_t0,* que serve para reiniciar o contador em 0 e é uma das saídas do decodificador de sinais de controle.

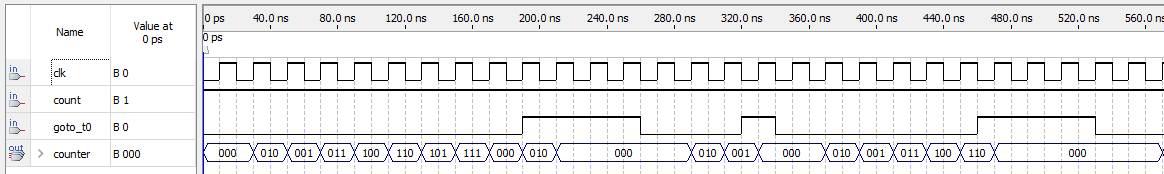
c0

c1

c2

goto\_t0

**Figura 4: decodificador das instruções do Neander.**

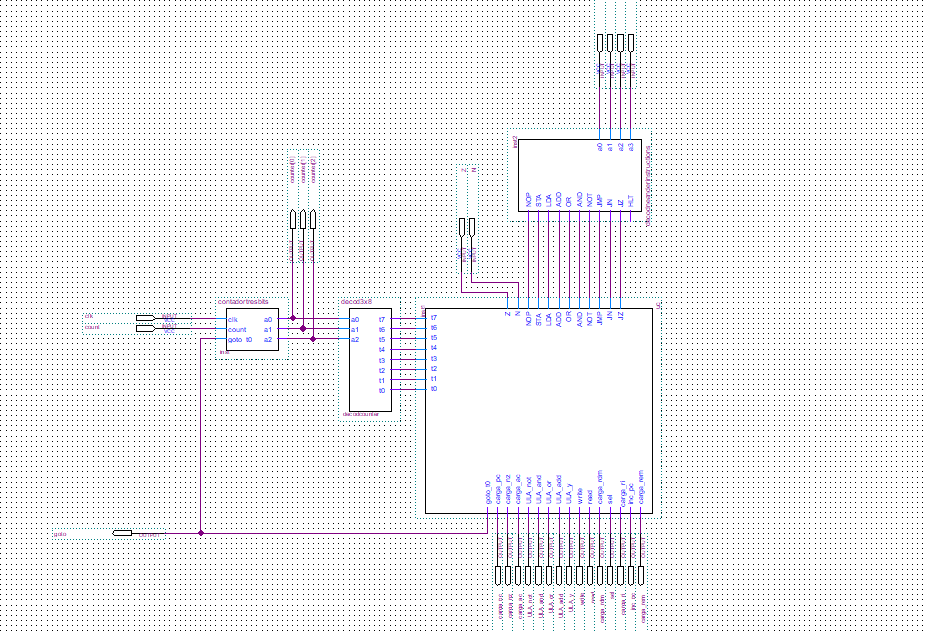
A seguir, a simulação desse contador:

**Figura 5: simulação do contador de três bits, com o *goto\_t0*.**

**Decodificador dos sinais de controle**

Serve para dizer qual sinal de controle deverá ser ativado no determinado ciclo de clock. Recebe como parâmetros o decodificador de instruções do Neander, o contador de três bits e, adicionalmente, os sinais de N e Z, que indicam quando o operando a ULA é negativo ou zero, respectivamente. A figura 6 a seguir mostra esse decodificador com as suas entradas.

Note que há um componente adicional entre o contador e o decodificador de sinais. Foi optado por colocar um decodificador 3x8 para o valor do contador, para trabalhar com menos fios dentro do decodificador de sinais.



Saída de teste

Saídas de teste

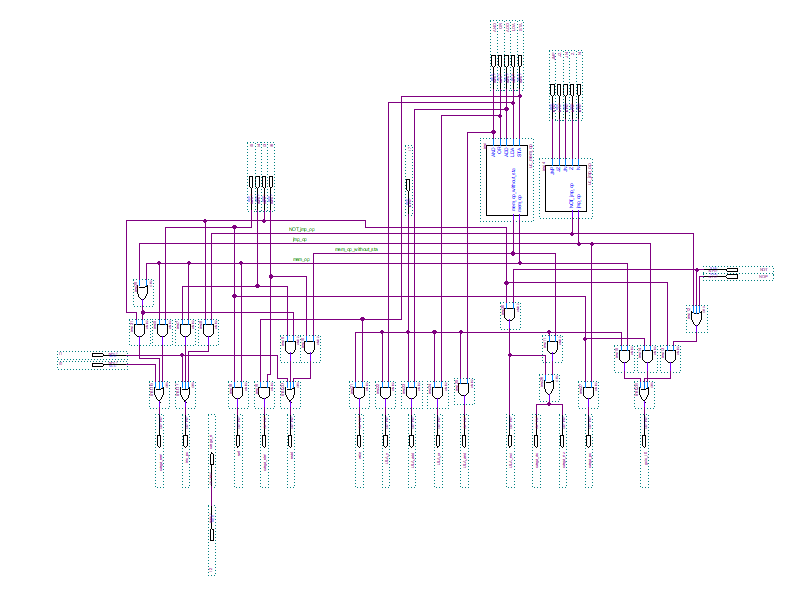
Decodificador de sinais

Contador

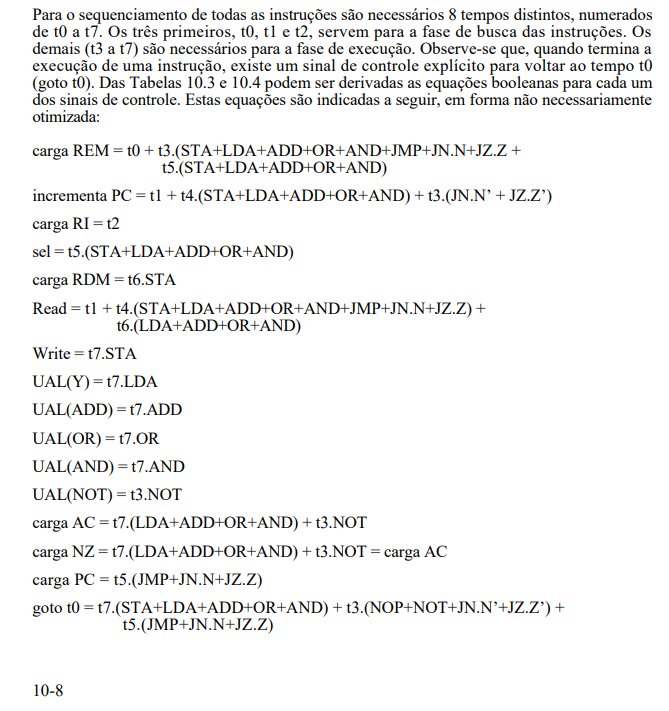
Decodificador de instruções

**Figura 6: estrutura da unidade de controle completa**

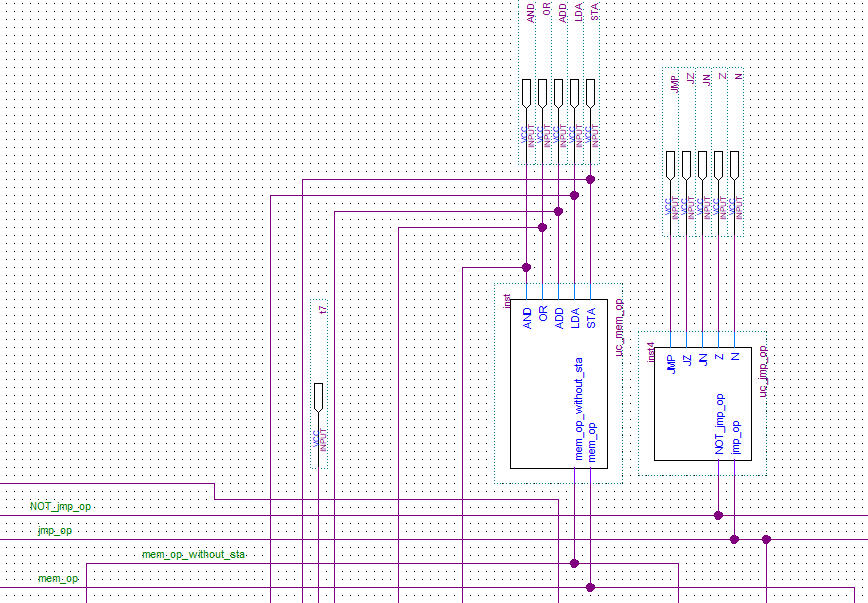
A seguir, na figura 7, o diagrama esquemático do decodificador de sinais. Esse diagrama foi feito com base no livro do professor Weber, como pode-se ver na figura 8. Além disso, algumas simplificações nos sinais foram feitas para diminuir mais a quantidade de fios no circuito. Essas simplificações estarão destacadas nas imagens.



**Figura 7: diagrama esquemático do decodificador de instruções do Neander**

**Figura 8: funções para cada sinais de controle.**

A seguir, as simplificações no circuito. As simplificações estão nos dois pequenos decodificadores presentes na parte superior do circuito, conforme mostra a figura 9.

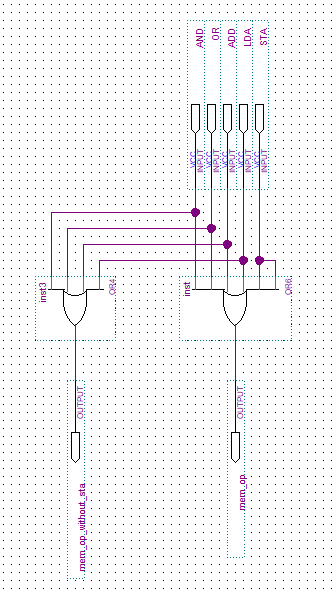


Decodificador de operações de desvio

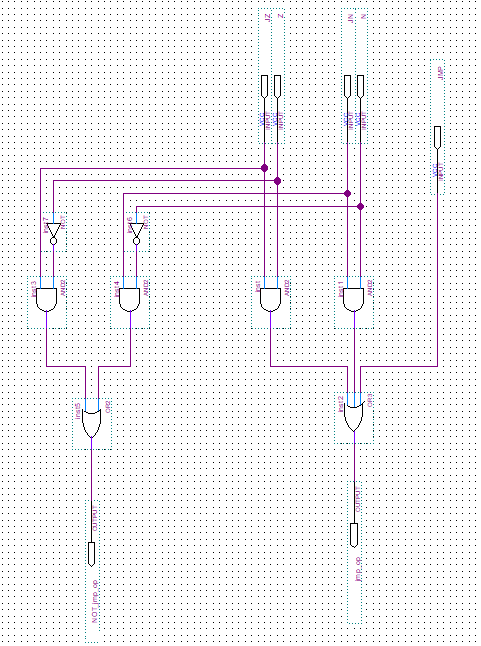
Decodificador de operações de memória e ULA

**Figura 9: parte superior do decodificador de sinais citada.**

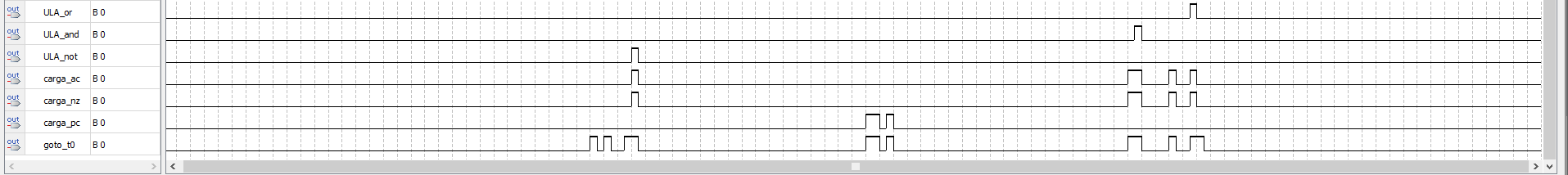
Nela, há dois decodificadores. Um relacionado as operações que lidam com a memória e a ULA (não estando presente a função NOT), e outro relacionado as operações de desvio.

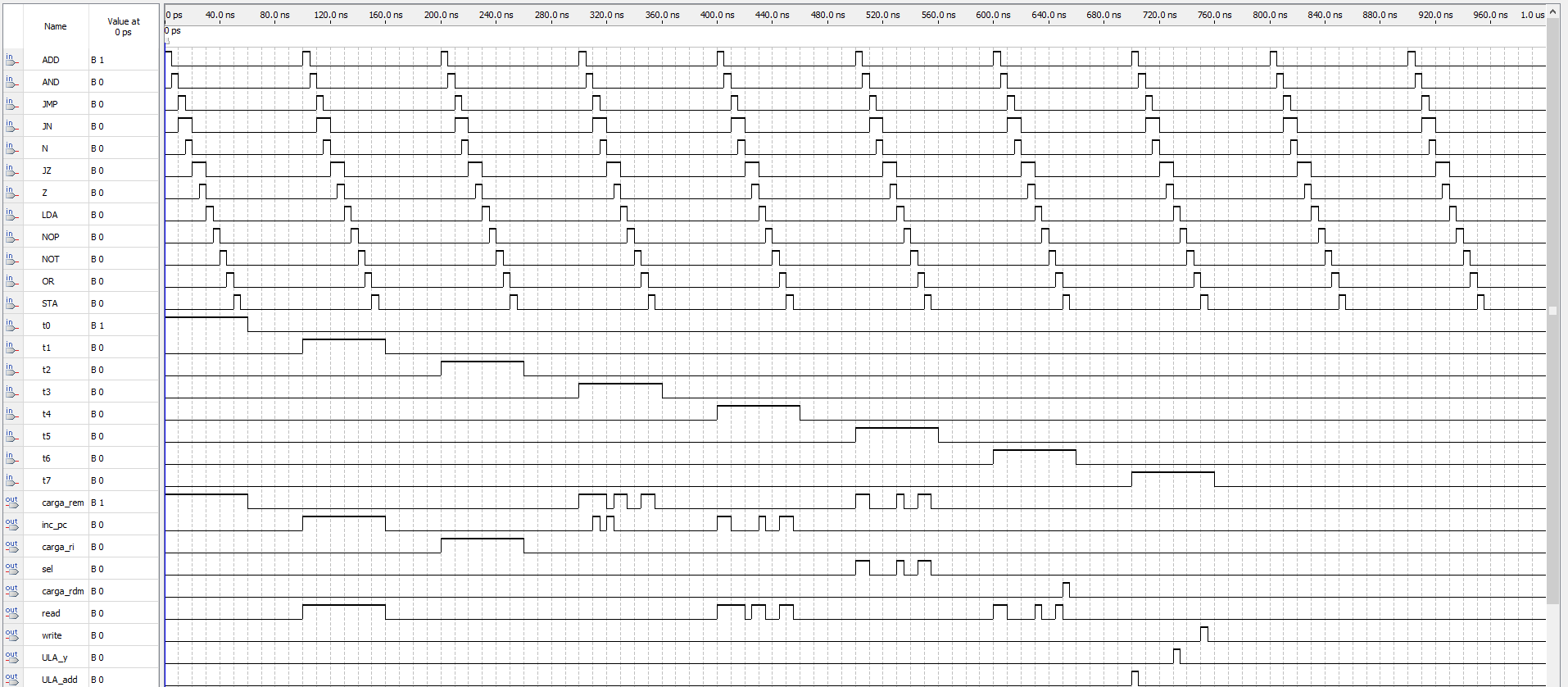
****

**Figura 10: decodificador das operações de memória e ULA.**

****

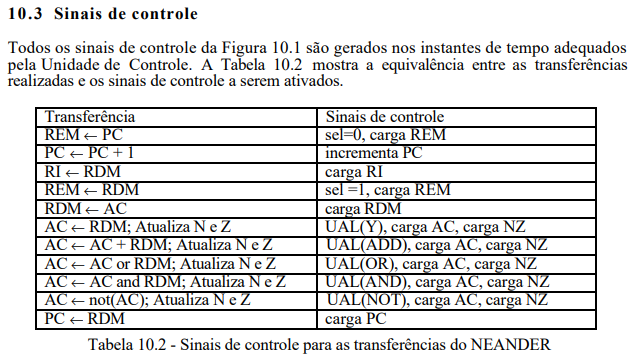
**Figura 11: decodificador das operações de desvio.**

**Conclusão:**

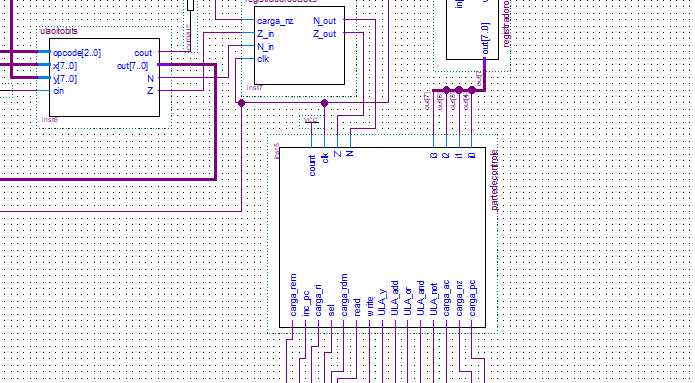
A seguir, toda a simulação da unidade de controle, na figura 12:

**Figura 12: simulação completa da UC.**

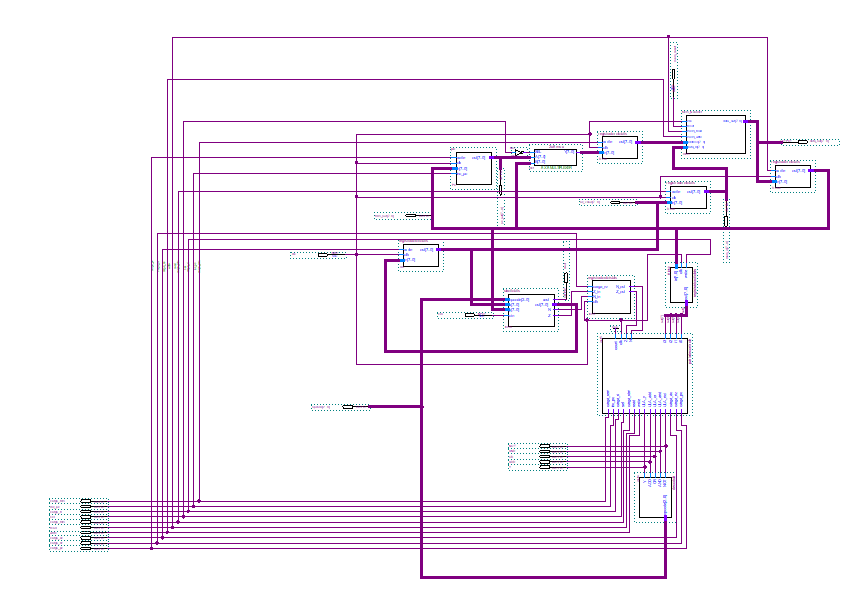
Lembrando que todos os sinais de controle foram baseados na tabela 10.2 do livro do professor Weber, como mostra a figura 13 a seguir:



**Figura 13: tabela de sinais de controle do Neander**

Com isso, temos nossa parte de controle feita e funcionando. A seguir, duas figuras que mostram onde essa parte se encaixa no *datapath* do Neander.

**Figura 13: parte de controle conectada ao resto do Neander.**

****

**Figura 14: Neander e seu *datapath* completo e funcional.**

**Dificuldade encontrada no desenvolvimento do decodificador de sinais:**

Ao desenvolver a parte de controle, quando o teste do decodificador de sinais estava sendo feito, foi percebido que quando o sinal de *goto\_t0* era ativado, o contador retornava imediatamente ao *t0,* não realizando as outras operações do tempo do estado atual do contador.

Após muito tempo, foi constado que a entrada *goto\_t0* não estava conectada a um Flip-Flop do tipo D (DFF) antes de ser conectado a porta CLRN do DFF. Então, quando o *goto\_t0* era ativado, imediatamente o contador era zerado e não dava tempo de ativar os outros sinais, visto que estes seriam ativados na próxima subida do Clock. Bastou conectar a entrada em um DFF e o decodificador de sinais funcionou corretamente.

Isso mostrou o quão importante é a barreira temporal que os Flip-Flops proporcionam e o quão estável um circuito síncrono é, porque tudo vai acontecer exatamente como se espera, uma instrução após a outra e um sinal sendo ativado após o outro.