

Projeto Final

24 de abril de 2021

1. Descrição do circuito

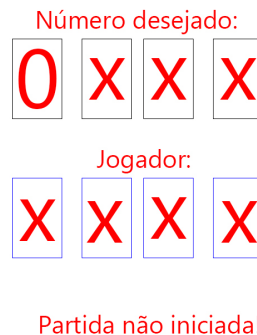
O circuito apresentado será um jogo individual de seleção de números.

2. Explicação do Jogo

O jogo começa com o computador gerando 4 números em hexadecimal. Após terminado a seleção, será fornecido uma sequência de números hexadecimais de 4 bits no total. Ganha o jogador que selecionar os números fornecidos de forma que fiquem iguais ao números gerados.

Mostrarei imagens hipotéticas para exemplificação do projeto.

Jogo inicializando:



Sequência fornecida pelo computador. Agora, o jogador precisa acertar a sequência:

Número desejado:
5 4 3 9

Jogador:
0 X X X

Partia em andamento!

Vitória, pois as sequências são iguais.

Número desejado:
5 4 3 9

Jogador:
5 4 3 9

VITÓRIA!

Derrota por sequências diferentes.

Número desejado:
5 4 3 9

Jogador:
5 1 X X

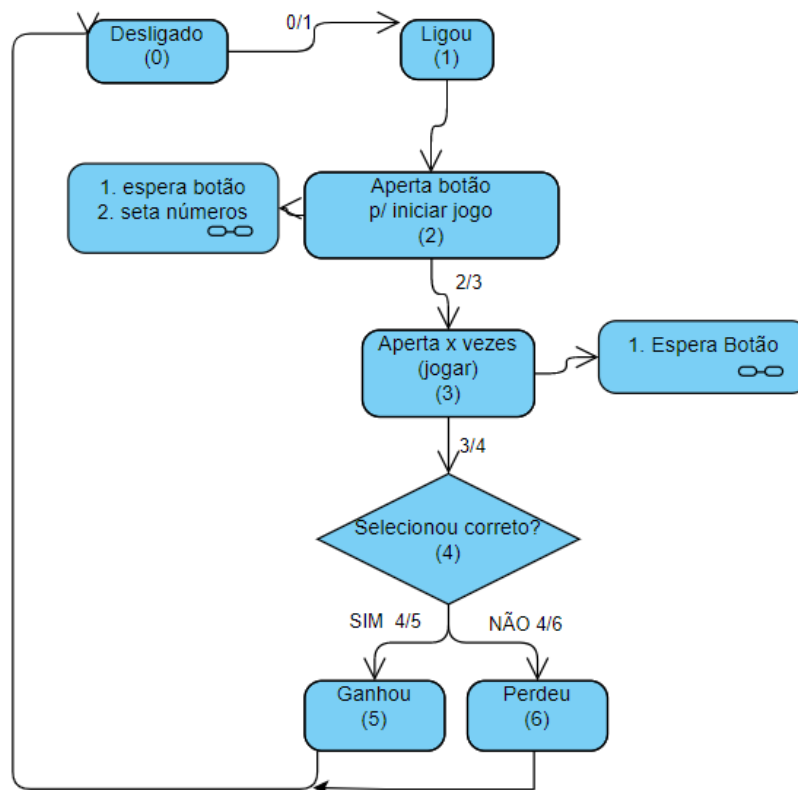
DERROTA :(

A inspiração para o jogo partiu de jogos de acertar "escalas". A principal ideia foi aumentar a dificuldade desse tipo de jogo. Por que acertar um número quando podemos acertar 4 ao mesmo tempo?

Agora apresentarei a máquina de estados, os blocos individuais e o "bloco principal" de funções do jogo. Vale lembrar que o entendimento dos blocos específicos pode ficar vago em sua subseção. Porém, ao apresentar o contexto onde ele se aplica, todas as dúvidas serão sanadas.

3. Máquina de Estados

Aqui a máquina de estados do jogo.

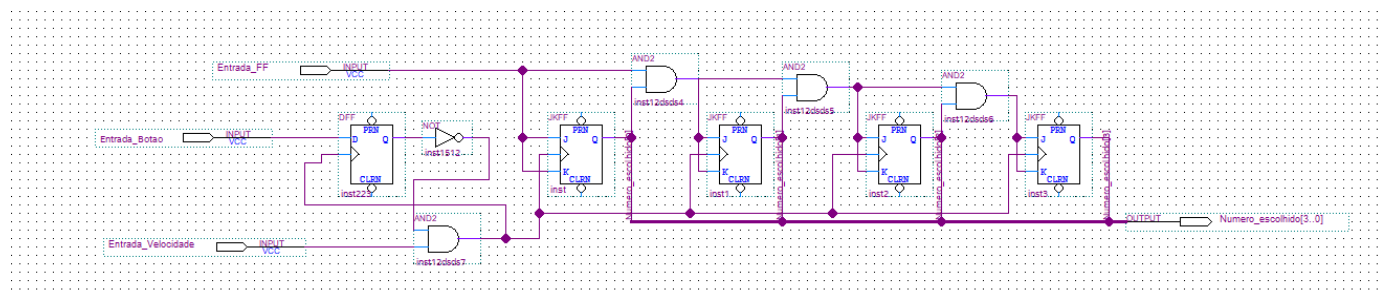


4. Bloco: Seletor 4bits

Esse é o bloco que fornece a sequência de números. Ele é um contador síncrono binário de 4 bits e existe um fator chave para o funcionamento desse bloco que pode não ser aparente a primeira vista.

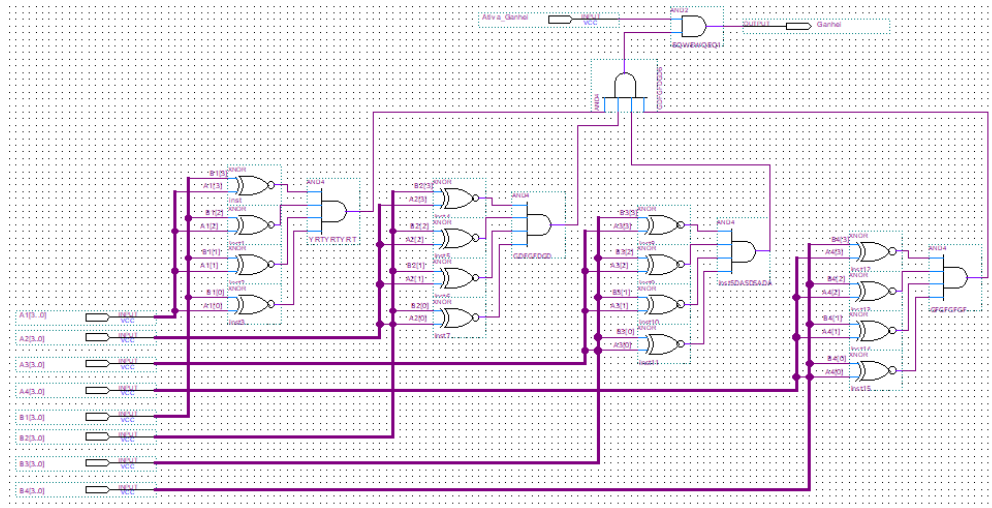
O Bloco tem 3 inputs: Entrada Velocidade, responsável pelo clock dos flip flops, Entrada Botão, responsável pelo sinal do botão(interacção do usuário) e Entrada FF, responsável por habilitar a função. Ele também possui 1 output chamado Número escolhido. Esse output, evidentemente, guarda o resultado da seleção.

O diferencial do Bloco fica no input do botão. Todos os botões funcionam como clocks no jogo que estamos tratando. A dúvida seria: como selecionar o número do contador? Para resolver isso, coloquei a saída do flip flop D inicial negada em uma and junto com o clock(entrada velocidade). A saída dessa and é utilizada para o clock do flip flop D. Partimos do princípio que a saída Q começará em 0 ao iniciar o programa e, ao ser negada, deixara a and "setada" para receber o clock. Quando o primeiro sinal de botão chegar, a saída Q virará 1, resultando em 0(pela negação) e trancando a porta and. Por fim, trancar a porta and corta a corrente clock no bloco e resulta em uma memória em um output do último número contado. Vale ressaltar a título de curiosidade que coloquei o nome "Entrada Velocidade" no clock para ficar mais amigável ao usuário, ou seja, facilitar a interpretação de que esse sinal é a alteração de velocidade que o contador conta(dificuldade principal do jogo).



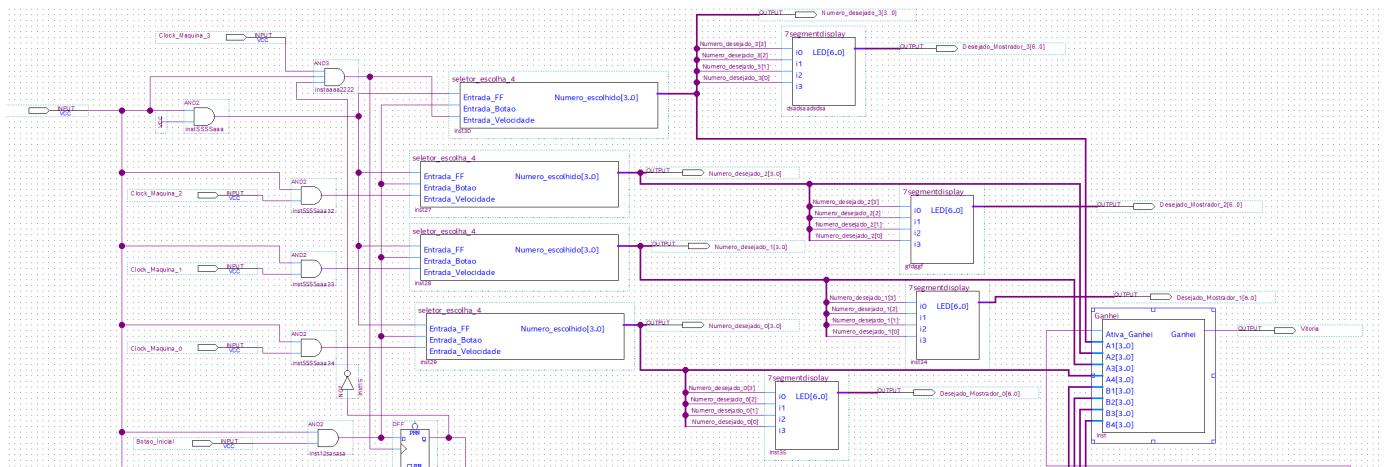
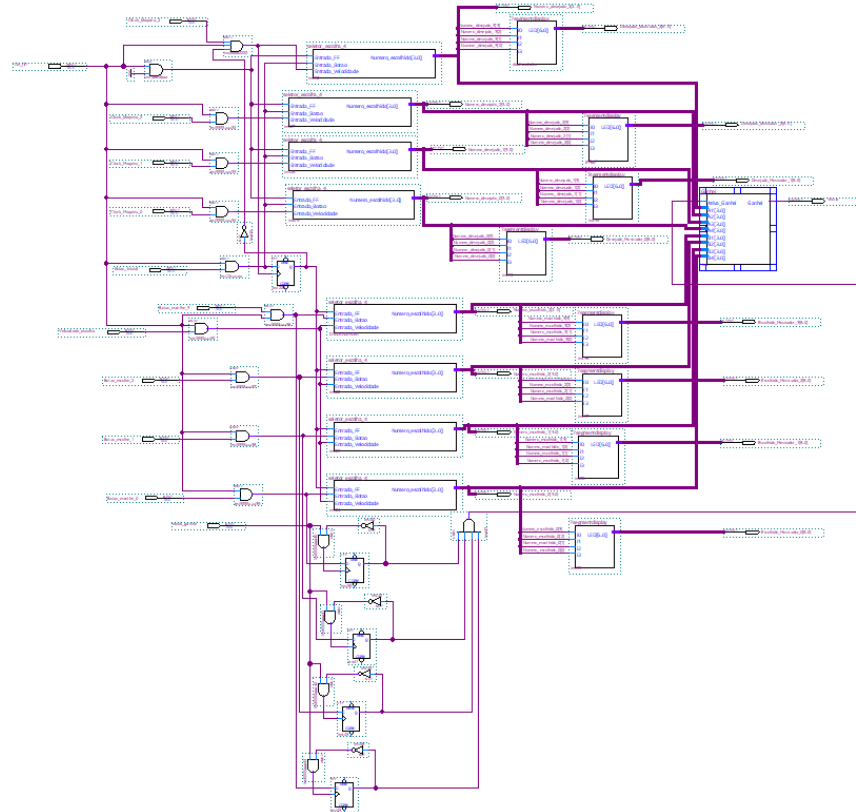
5. Bloco: Ganhei

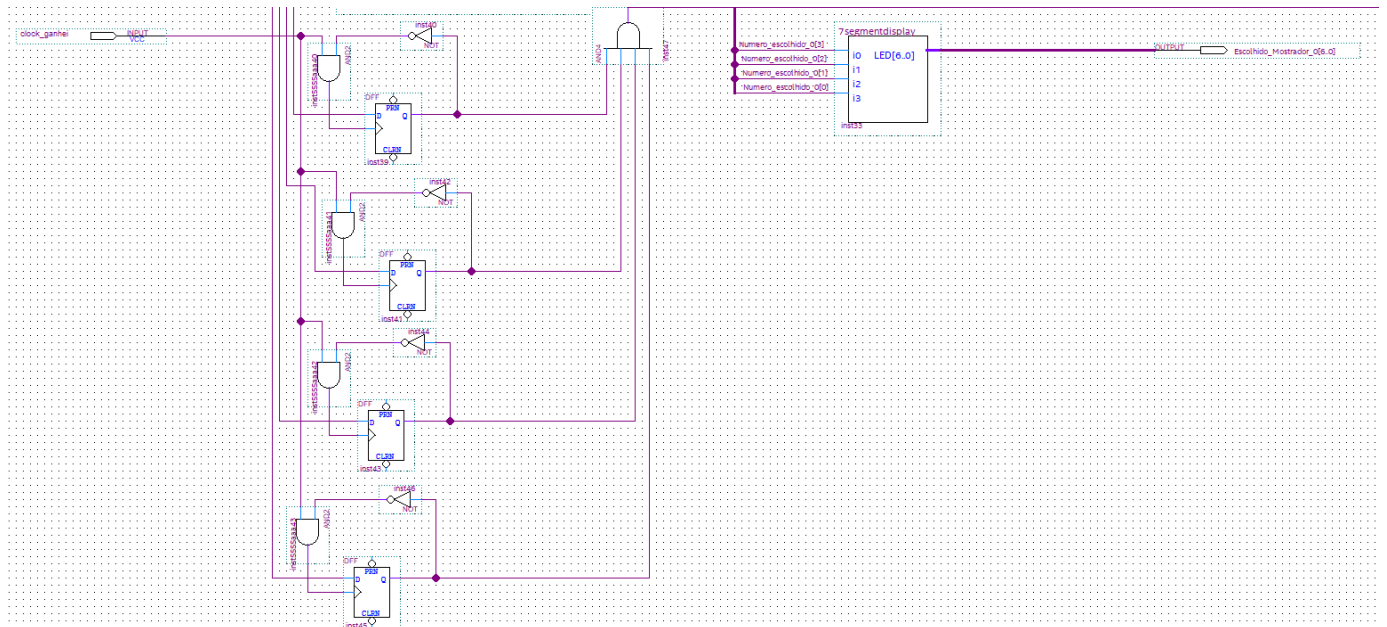
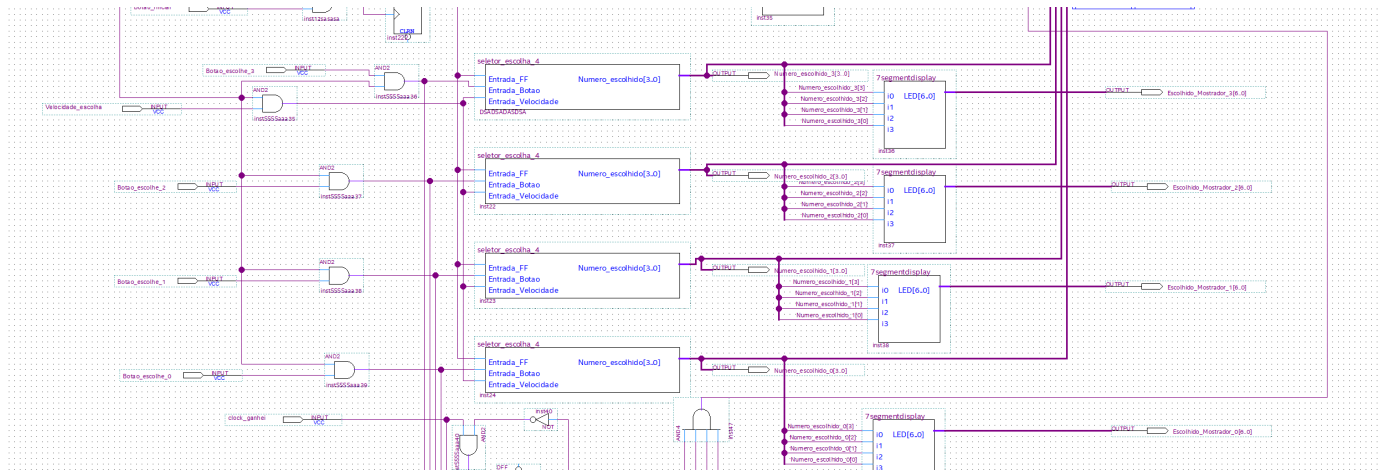
Esse bloco recebe os números selecionados pelo circuito e pelo jogador e compara para ver se o jogador ganhou o jogo. Se os números forem iguais, vitória = 1. Números Diferentes, Vitória = 0(derrota).



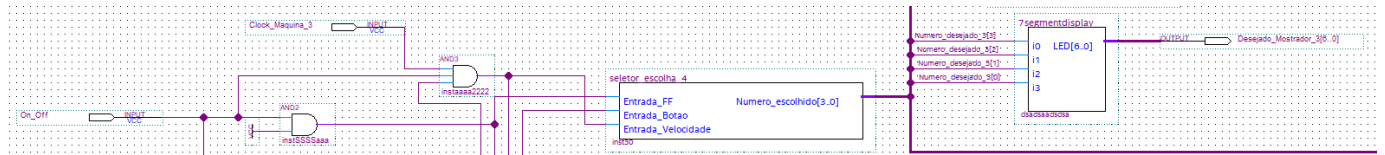
6. Desenho completo do Circuito

Colocarei primeiro as fotos completas do Circuito. Após, explicarei detalhadamente o caminho que ele percorre.



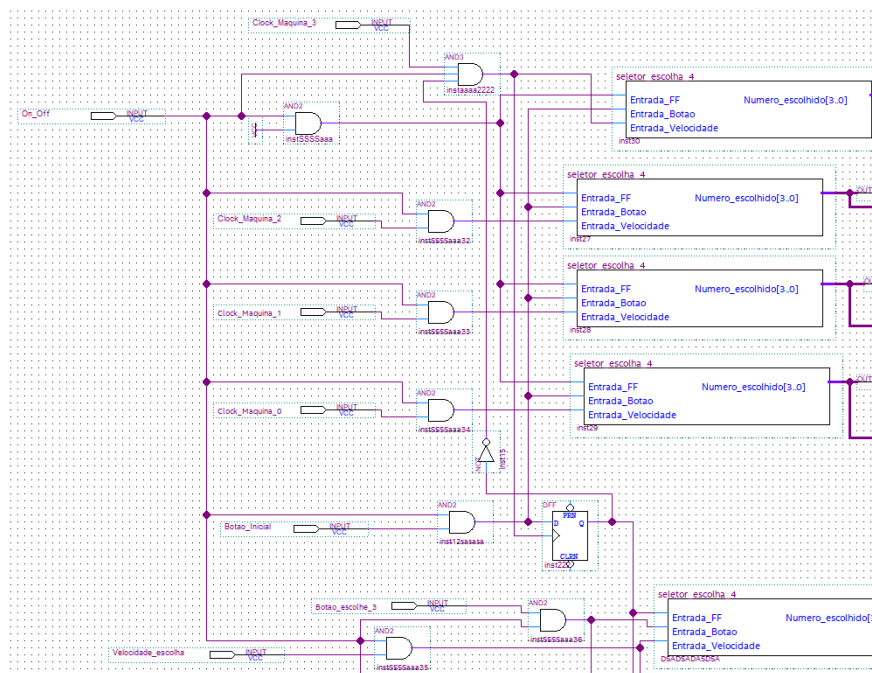


Aqui mostro o input On Off. Esse input de 1 bit define se uma máquina está ligada ou não. Todos os outros inputs do código tem ands com ele, ou seja, "as entradas só funcionam com o circuito ligado".

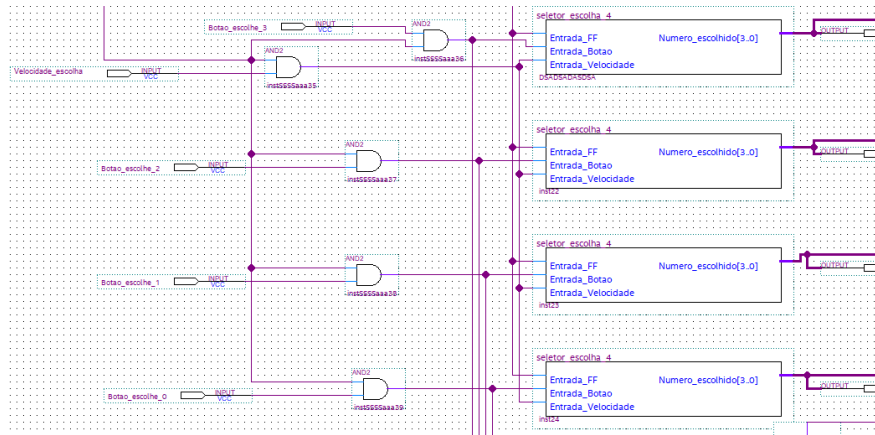


Aqui apresento os 4 primeiros seletores de 4 bits. Esses são os blocos que o circuito utilizará para gerar os números do jogo. Repare que existem uma entrada de clock diferente para cada seletor, pois, se todas fossem iguais, os números fornecidos por eles seriam também iguais. Cada seletor apresentará o seu número escolhido em um display de 7 segmentos.

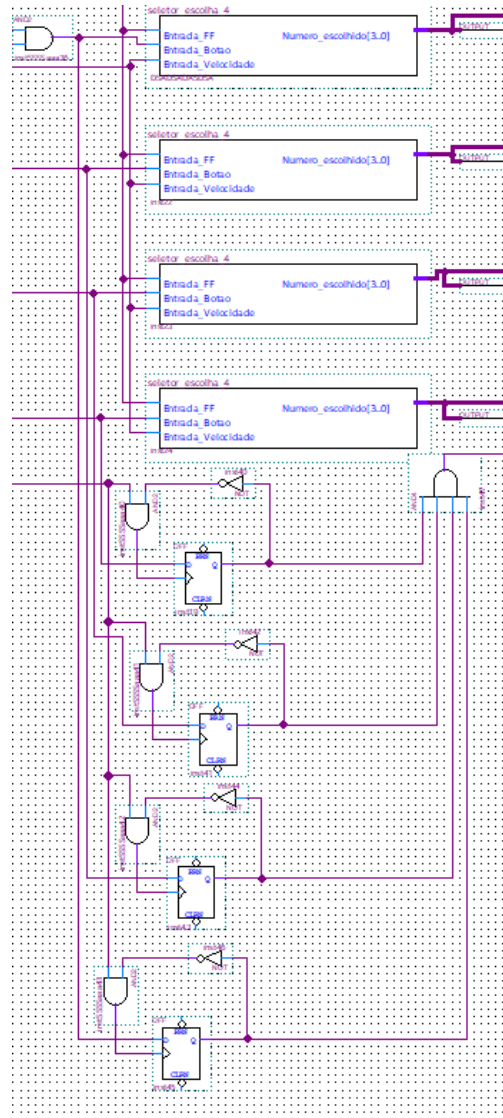
Analisarei também a função do input botão inicial. O jogo só começa a partir do momento que o botão inicial está ativo. O tempo de diferença entre a ativação de On Off e o pressionar de botão de botão inicial é o tempo que gerará números diferentes para o jogo. O botão inicial utiliza o mesmo sistema de flip flop D que os seletores de escolha. Isso é importante para que seja enviado um só pulso de escolha para cada "entrada botao" dos seletores. Vale lembrar também que é nesse flip flop que a parte de interação do usuário é "liberada para funcionar", ou seja, o flip flop ativa um pulso constante positivo para a segunda partes de seletores do código que mostrarei a seguir.



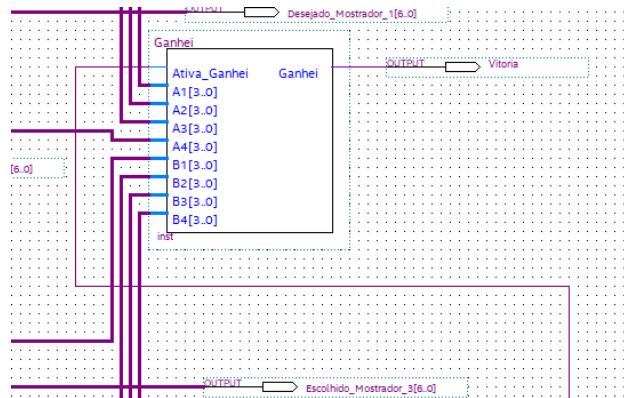
Aqui apresento os 4 últimos seletores de 4 bits. Todos os seletores possuem a mesma velocidade de escolha, ou seja, o mesmo clock. Isso acontece, pois os números precisam ser apresentados para o jogador de uma forma similar. Sequências de pulsos diferentes dificultariam demais o processo de escolha do número por parte do jogador. Cada um dos quatro seletores possui o seu próprio botão para selecionar o número desejado. Eu poderia ter implementado essa parte com um só botão, mas isso tiraria do jogador a possibilidade de escolha de números na ordem alternada. Quatro botões foi uma opção para facilitar o entendimento do usuário. Todas as entradas "Entrada FF", ou seja, os "enables" dos contadores binários estão sendo ativas pelo pulso constante positivo gerado pelo flip flop d ligado ao botão inicial. Essa é a conexão dos código que comentei a cima.



Aqui eu mostro flip flops D utilizando a mesma lógica já abordada diversas vezes no circuito. A função desses flip flops é captar e mandar positivo um sinal a partir de cada botão apertado no código. Se os 4 sinais forem captados em uma and e a porta resultar em sinal positivo, significa que todos números foram selecionados. A partir do momento que temos os números de cima e baixo selecionados, podemos compara-los na função ganhei para verificar se o usuário venceu ou não a partida.

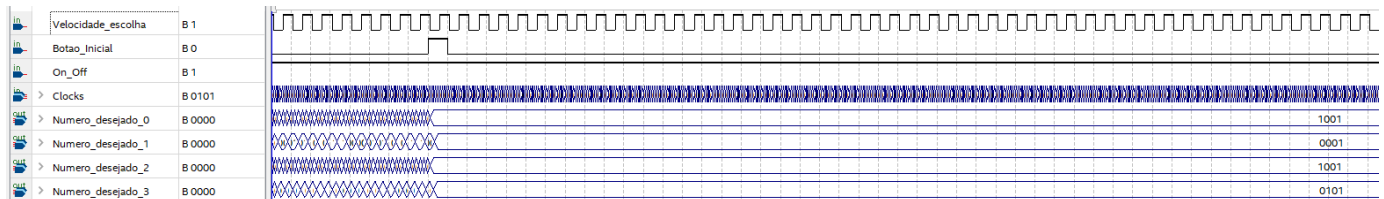


A saída da and apresentada é inserida na entrada "Ativa Ganhei" do bloco, para, como o nome já diz, ativar a função de teste de vitória.

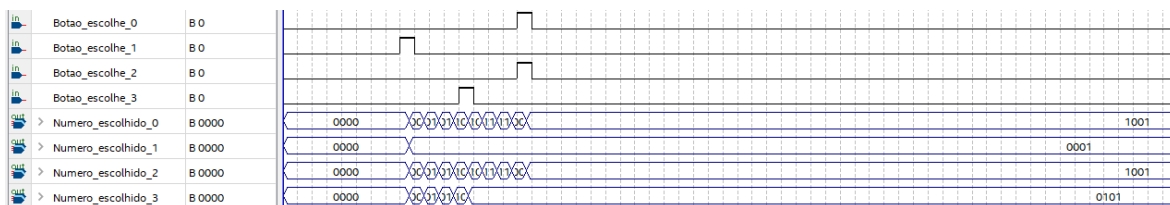


7. Simulação Funcional do Circuito

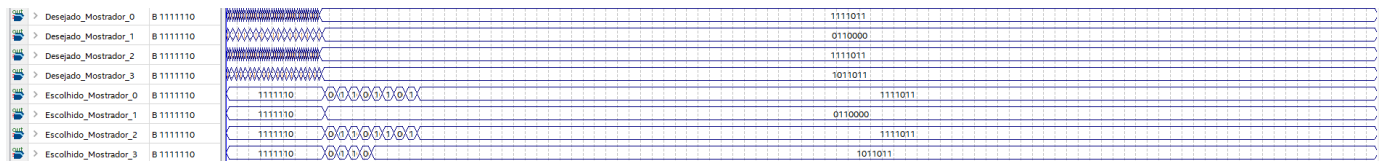
Aqui mostro o funcionamento do botão inicial. A partir do momento em que o jogador aperta o botão, os números são escolhidos pelo computador. "Clocks" é o grupo de inputs das velocidades de cada um dos seletores.



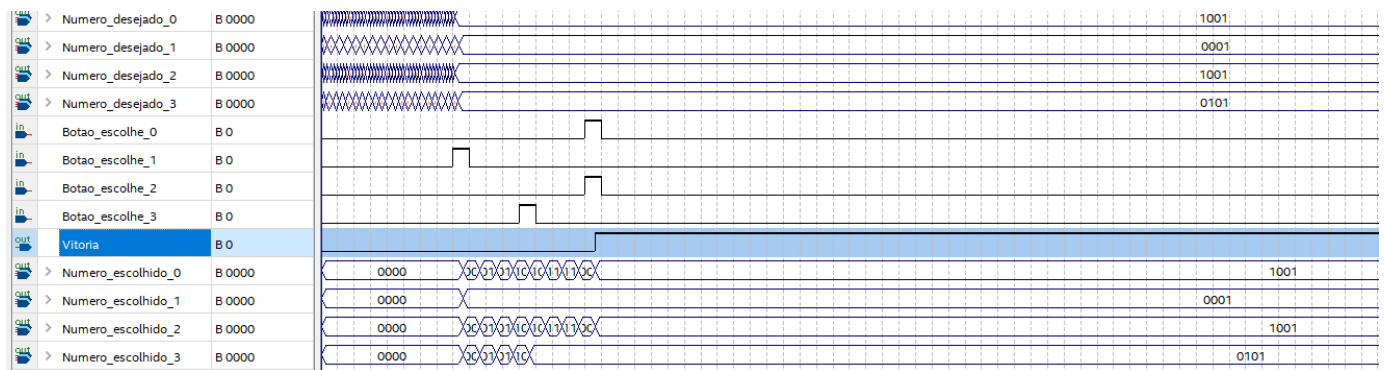
Aqui mostro o usuário apertando os 4 botões e os números selecionados aparecendo no output "número escolhido". Repare que a quantidade de números até a seleção do número pelo usuário é muito menor do que nos bits selecionados pelo computador. Isso se deve a dois motivos: (1) a sequência deve ser mais lenta para que o usuário consiga interpreta-lá e interrompe-lá e (2) sequências mais rápidas tem a possibilidade de gerar números mais distantes mesmo com tempos de seleção próximos. Repare também que os botões são todos apertados após o usuário ativar o botão inicial, ou seja, ativar o jogo.



Aqui uma foto dos displays apresentados para o usuário. Cada um representa um dos seletores a cima. Os selecionados pelo circuito são chamados "desejado" e os selecionados pelo jogador são chamados "escolhido".



Por fim, mostro o bit vitória amostrando que o jogador venceu a partida.



4. Conclusão sobre a atividade

Mais uma batalha travada com o quartus II com sucesso. Enfim chegamos ao trabalho final. Chegar ao fim desse projeto mostra que o semestre foi bem sucedido. Não digo isso pela dificuldade do projeto proposto, mas por saber que consegui unir os recursos aprendidos para aplicar um circuito totalmente idealizado por mim. Confesso que todas as horas "batendo cabeça" com o Quartus realmente se fizeram valer para o projeto final. Estou muito satisfeito com o que produzi e ansioso pelo que virá pela frente.