



**UNIVERSIDADE
FEDERAL DO CARIRI**

**UNIVERSIDADE FEDERAL DO CARIRI
CENTRO DE CIÊNCIAS E TECNOLOGIA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**Projeto 02 - Guess The Number Xtreme
Professor: Ramon Santos Nepomuceno**

**Wanderson Faustino Patricio
Francisco Anderson Maciel Cruz**

Juazeiro do Norte, 26 de outubro de 2023

Sumário

1	Explicação do jogo	3
2	Circuitos Utilizados	4
2.1	Flip-Flop tipo D	4
2.2	Registrador	5
2.3	Unidades	6
2.4	Dezenas	7
2.5	Contador Regressivo	8
2.6	Circuito de preset	9
2.7	Cronômetro	10
2.8	Relógios	11
2.9	Negativo	12
2.10	Circuito Somador	13
2.11	Comparador	14
2.12	Circuito Próximo	15
2.13	Core	16
2.14	Placar	18
3	Conexões Adicionais no Jogo Principal	19

1 Explicação do jogo

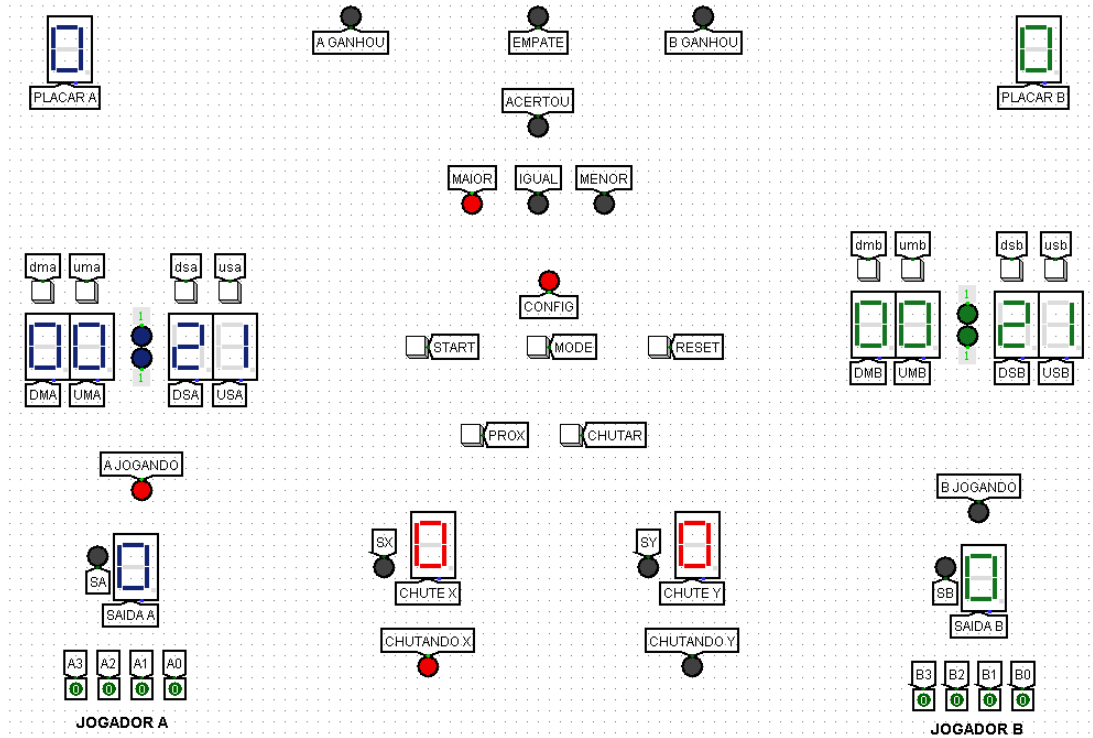


Figura 1: Visão do jogo

Guess the Number Xtreme é um jogo de adivinhação de números. O jogo é jogado por dois jogadores (A e B) que devem adivinhar as coordenadas de um ponto em um plano cartesiano.

Os jogadores alternam-se entre as rodadas chutando as coordenadas x e y (Com seus respectivos sinais), os valores de chutes são números inteiros variando entre -7 e 7 , sendo ao total 15 possíveis chutes para x e y , resultando em 225 chutes de pontos.

A cada chute dado pelos jogadores é informado se a soma dos valores das coordenadas é maior, menor ou igual a soma das coordenadas reais do ponto a ser descoberto. Caso o jogador descubra os dois valores, o botão **acertou** é aceso, e adicionado um ponto ao placar do jogador. A cada acerto um novo ponto é gerado aleatoriamente, e os jogadores podem continuar adivinhando. O jogo termina quando o tempo dos dois jogadores acaba, ou quando um deles alcança 15 pontos. O jogo então informa se alguém ganhou ou se houve empate.

Para informar o seu chute, o jogador deve apertar o botão **PROX** para confirmar o chute, ou o botão **CHUTAR** para apenas passar o chute para o display, podendo ser alterado até o botão **PROX** ser apertado.

O tempo dos jogadores pode ser definido ao início do jogo através do botão **Mode** (podendo alterar de 1s até 59m59s). Para iniciar o jogo basta sair do modo de configuração e apertar o botão **start**. Ao final do jogo basta apertar o botão **reset** para zerar o jogo e poder jogar novamente.

2 Circuitos Utilizados

2.1 Flip-Flop tipo D

Para o nosso jogo precisaremos de elementos de memória, visto que precisaremos guardar os chutes dos jogadores, o placar de cada um, o tempo atual dos jogadores, o ponto gerado no interior do circuito **CORE**, etc. Utilizaremos o componente flip-flop do tipo D para isso.

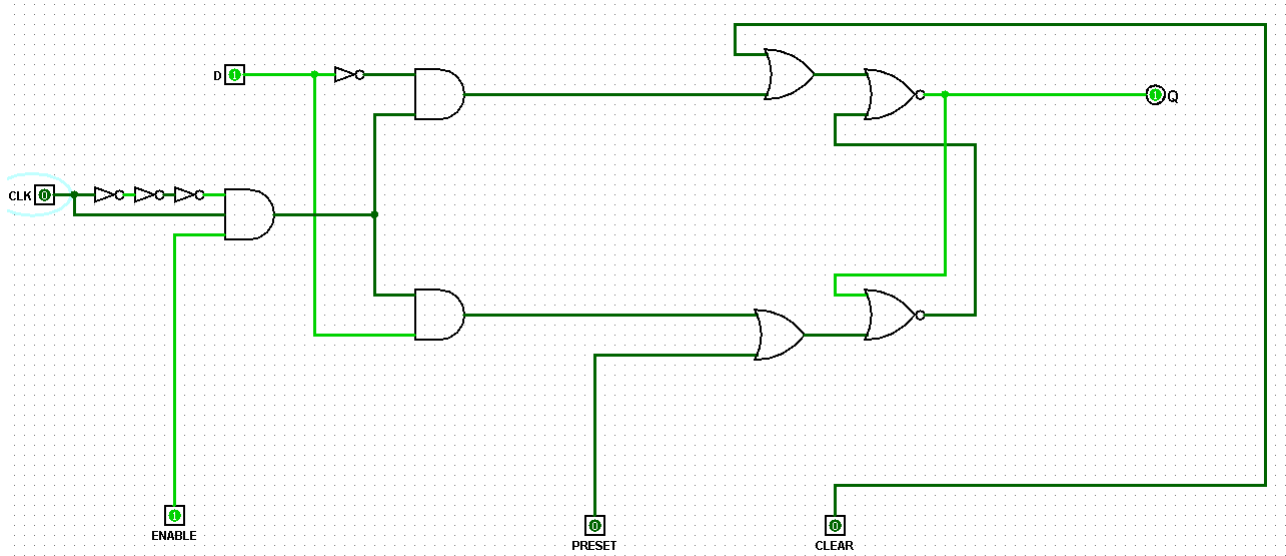


Figura 2: Flip-flop tipo D

O funcionamento do flip-flop tipo D se baseia na borda de subida do clock. Sempre que a entrada de clock (**CLK**) sai de nível baixo para alto, o valor que está em **D** é transmitido para a saída **Q**. Esse valor ficará guardado até a próxima subida do clock, independentemente de haver ou não mudanças no valor de **D**. Todavia, se houver mudanças em **D** no momento de subida de **CLK** o valor em **Q** mudará.

Foi adicionado também um botão de **enable** que permite ou não passar a entrada do clock, um botão de **preset**, que altera o valor em **Q** para 1 independente do restante do circuito, e um botão de **clear**, que faz o valor em **Q** tornar-se 0.

Quando esse circuito está habilitado o valor transmitido a **Q** é guardado até a próxima borda de subida do clock, sendo este o nosso elemento de memória de valores de 1 bit.

2.2 Registrador

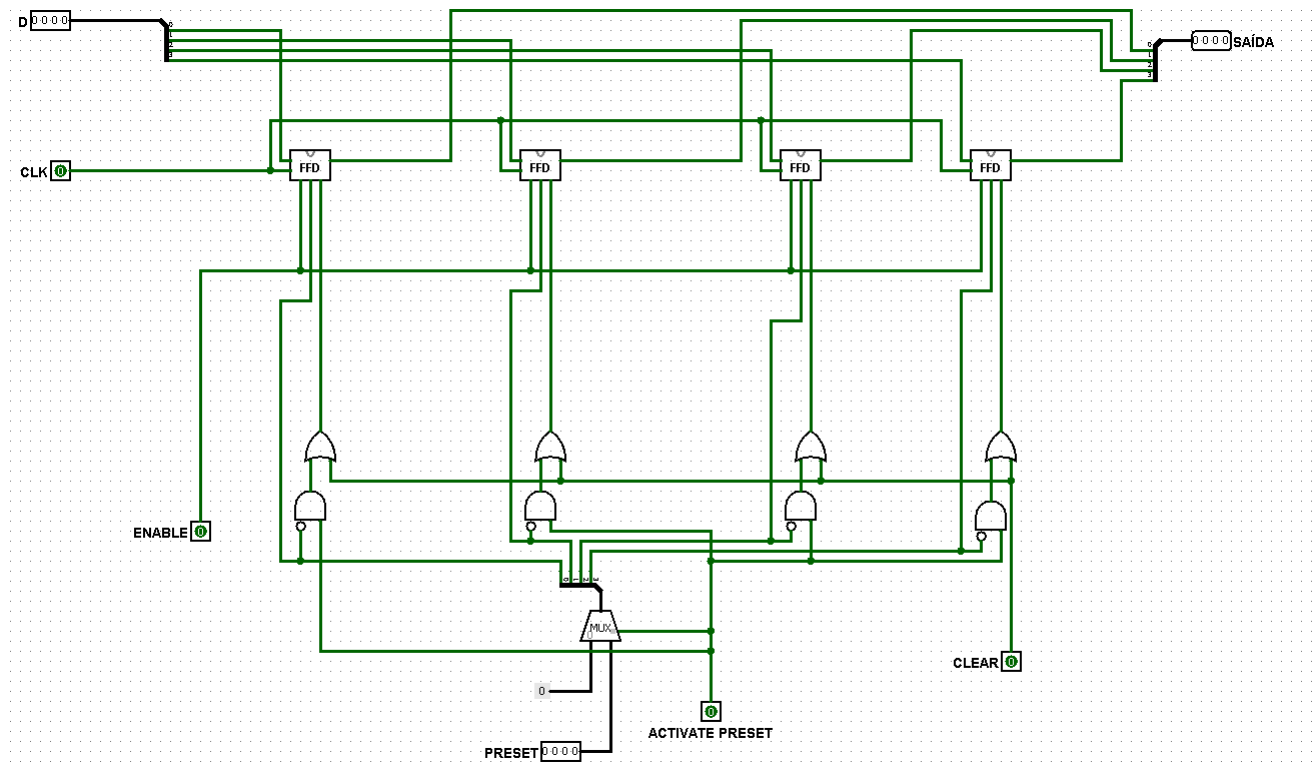


Figura 3: Registrador de 4 bits

O registrador é uma extensão dos elementos de memória de um bit. Para sua criação precisamos associar vários flip-flops tipo D em série conectados a mesma entrada de clock. Ao invés desse circuito armazenar apenas um bit, ele é capaz de salvar vários bits ao mesmo tempo. No nosso registrador utilizaremos 4 bits de memória.

A entrada **D** de 4 bits é distribuída pelos 4 flip-flops tipo D, todos conectados a mesma entrada de clock (**CLK**), suas saídas são então redistribuídas novamente para uma saída de 4 bits. A cada borda de subida do clock o valor de **D** é guardado em **Saída**.

A entrada **Enable** possui o mesmo funcionamento do flip-flop. Ela é conectada simultaneamente a todas as entradas de **Enable** dos flip-flops. Analogamente, a entrada **Clear** passa o número 0000 para a saída, independente do resto.

A grande diferença entre o registrador do logisim e o criado por nós é a entrada **Activate Preset**. É possível escolher um valor diferente de 0000 para ir a saída, independente do resto do circuito. caso o botão esteja desativado, as entradas de **preset** dos flip-flops estará sempre em 0, caso contrário, o valor da saída será a mesma do **preset**. Se o valor do bit de preset for 1, o botão de **preset** do flip-flop é ativado, se for 0 é ativado o clear do mesmo.

2.3 Unidades

O circuito funciona de maneira semelhante a um contador binário de 4 bits, operando como um cronômetro regressivo. A cada pulso de clock, o próximo estado do circuito é ativado. Os registradores de 4 bits armazenam o valor atual da contagem. Quando o valor atinge 0, o circuito de controle entra em ação, reiniciando o contador de volta a 9 e enviando um clock para a próxima estrutura que simula as dezenas.

Essa funcionalidade é possível graças a uma tabela de estados que controla as transições entre os valores de contagem e garante que o contador funcione como um cronômetro regressivo.

E3	E2	E1	E0	E3	E2	E1	E0	S
0	0	0	0	1	0	0	1	0
0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	1	0
0	0	1	1	0	0	1	0	0
0	1	0	0	0	0	1	1	0
0	1	0	1	0	1	0	0	0
0	1	1	0	0	1	0	1	0
0	1	1	1	0	1	1	0	0
1	0	0	0	0	1	1	1	0
1	0	0	1	1	0	0	0	1
1	0	1	0	1	0	0	1	1
1	0	1	1	1	0	0	1	1
1	1	0	0	1	0	0	1	1
1	1	0	1	1	0	0	1	1
1	1	1	0	1	0	0	1	1
1	1	1	1	1	0	0	1	1

Figura 4: tabela das unidades

Em resumo, o circuito mantém a contagem decrescente regularmente. No entanto, quando atinge 0, ele volta a 9, criando um ciclo contínuo de 9 a 0 e de volta a 9.

Após a análise da tabela de estados, o circuito segue a seguir:

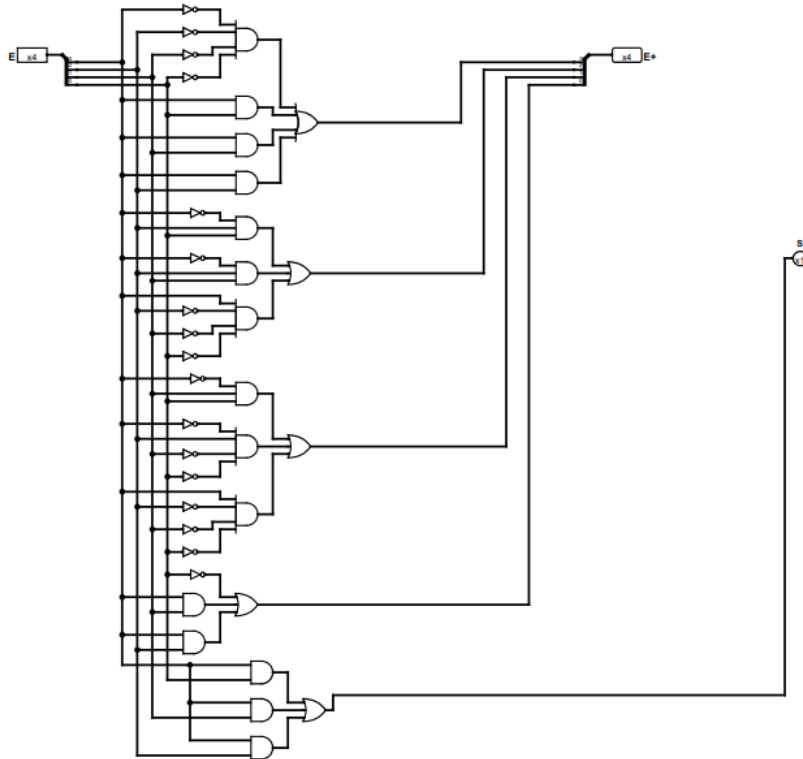


Figura 5: circuito das unidades

2.4 Dezenas

Este circuito é projetado com base em um contador binário de 4 bits, mas com cronometragem regressiva. A cada pulso de clock, o circuito avança para o próximo estado, seguindo uma sequência de contagem decrescente. Quando o contador atinge o estado "0000," em vez de reiniciar a partir de "0001," ele retrocede diretamente para o estado "0101."

Isso é possível devido a uma tabela de estados que determina as transições entre os valores do contador. A tabela de estados garante que o contador siga um ciclo de contagem regressiva consistente. Portanto, a sequência de contagem opera da seguinte maneira:

E3	E2	E1	E0	E_3	E_2	E_1	E_0	S
0	0	0	0	0	1	0	1	0
0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	1	0
0	0	1	1	0	0	1	0	0
0	1	0	0	0	0	1	1	0
0	1	0	1	0	1	0	0	1
0	1	1	0	0	1	0	1	1
0	1	1	1	0	1	0	1	1
1	0	0	0	0	1	0	1	1
1	0	0	1	0	1	0	1	1
1	0	1	0	0	1	0	1	1
1	0	1	1	0	1	0	1	1
1	1	0	0	0	1	0	1	1
1	1	0	1	0	1	0	1	1
1	1	1	0	0	1	0	1	1
1	1	1	1	0	1	0	1	1

Figura 6: tabela das dezenas

Em resumo, o circuito mantém a contagem decrescente regularmente. No entanto, quando atinge 0, ele volta a 5, criando um ciclo contínuo de 5 a 0 e de volta a 5. Após a análise da tabela de estados, o circuito segue a seguir:

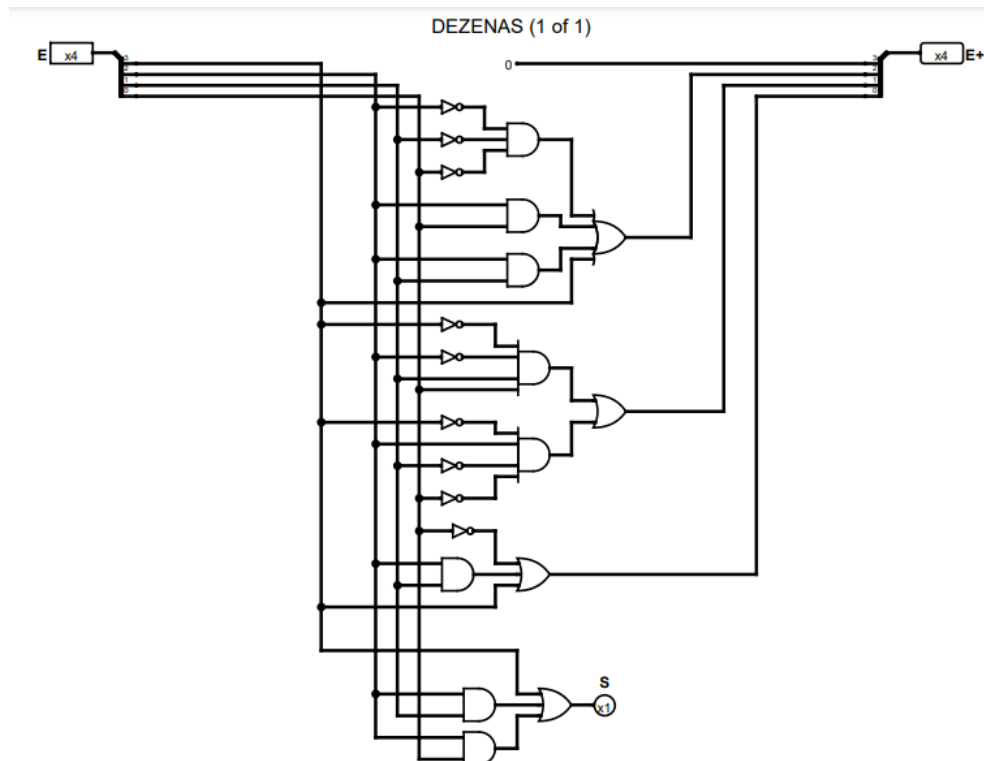


Figura 7: circuitos das dezenas

2.5 Contador Regressivo

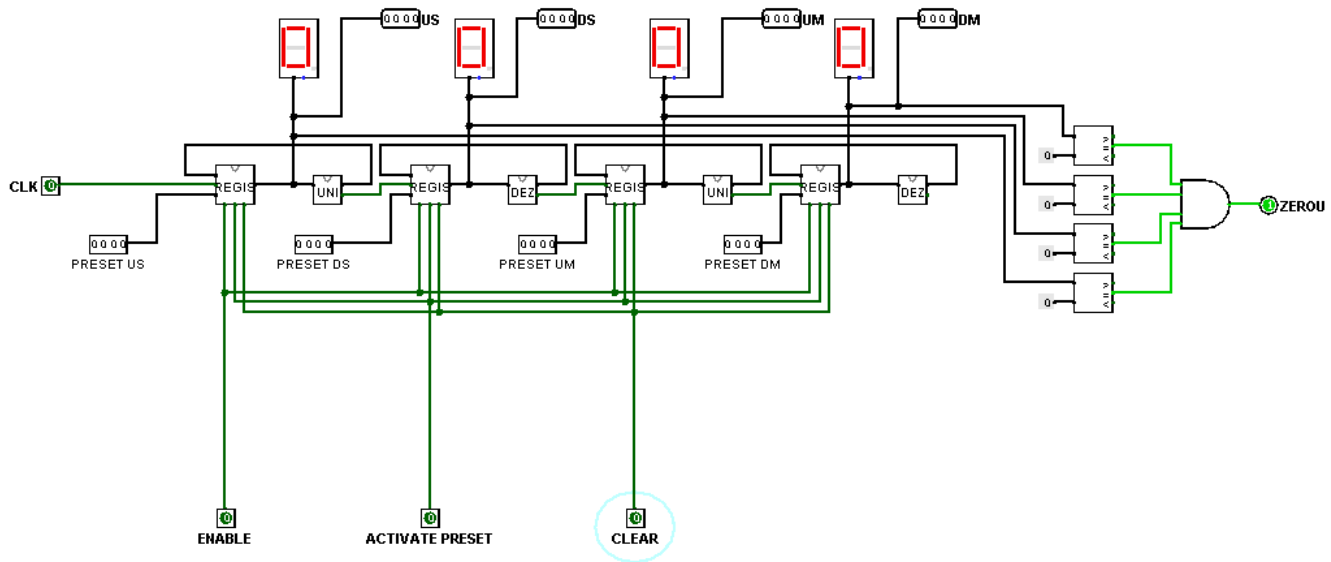


Figura 8: Contador Regressivo

Após adquirirmos os circuitos das unidades e dezenas, juntamente com os registradores, podemos conectá-los de maneira a criar uma simulação de contagem regressiva. Isso nos permite, inicialmente, decrementar o registrador das unidades de segundos, seguido pelas dezenas de segundos, unidades de minutos e, por fim, dezenas de minutos. Com essa configuração, podemos usar o botão "preset" para inserir um valor específico no registrador e, através dos pulsos, iniciar a contagem regressiva. Ao combinar esses componentes, conseguimos criar um intervalo de contagem de 59 minutos e 59 segundos.

Neste circuito, os botões "enable" e "clear" foram conectados a todos os registradores de forma que o "enable" ative o circuito, enquanto o "clear" zera os valores dos registradores, restabelecendo-os para 0.

No final do circuito, temos quatro comparadores que ativam uma porta AND quando todos os valores atingem zero, possibilitando, assim, encerrar o jogo posteriormente.

2.6 Circuito de preset

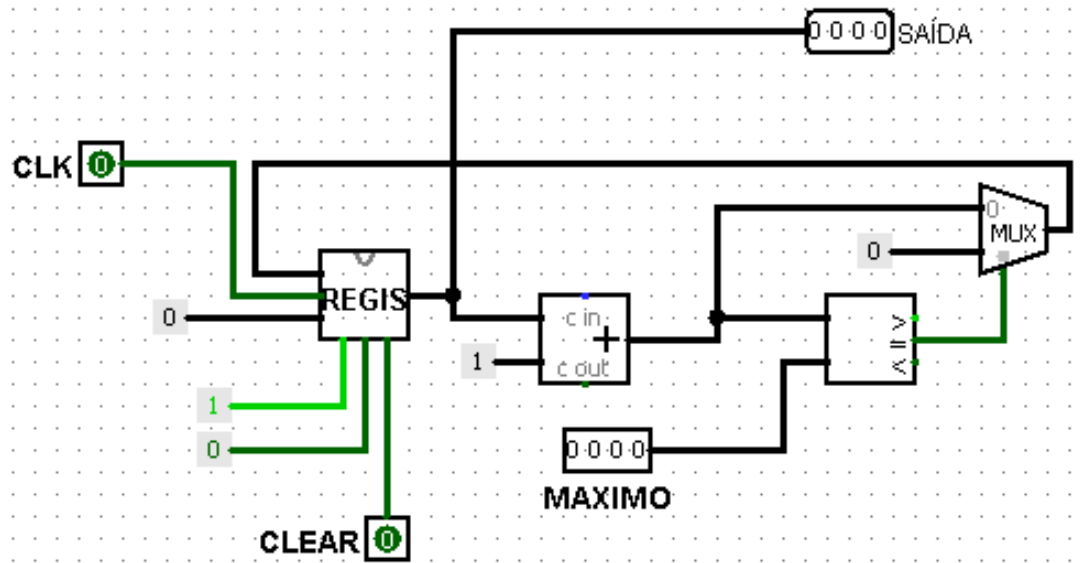


Figura 9: Circuito de Preset

O circuito de preset foi pensado para auxiliar na escolha do tempo inicial do cronômetro.

Para criar o circuito de preset basta utilizar um registrador com valor de preset igual a 0, **enable** sempre ligado, e **Activate Preset** sempre desligado. A entrada de **clear** é ligada a um botão para que seja possível zerar o **preset** caso seja necessário.

O funcionamento do circuito baseia-se em somar 1 ao estado atual a cada subida de clock. Para isso, a saída do registrador é conectada a um somador, que tem sua outra entrada conectada à uma constante 1. Essa saída é somada e entra na entrada **D** do registrador. Como a saída somada a 1 entra novamente no registrador, no próximo clock, o valor de saída será o valor atual mais um.

Todavia, é possível adicionar um limite para o valor máximo que o circuito pode alcançar. Quando a saída for igual ao máximo estabelecido, o seletor do multiplexador é ativado e passado zero à entrada **D** do registrador, reiniciando a contagem.

2.7 Cronômetro

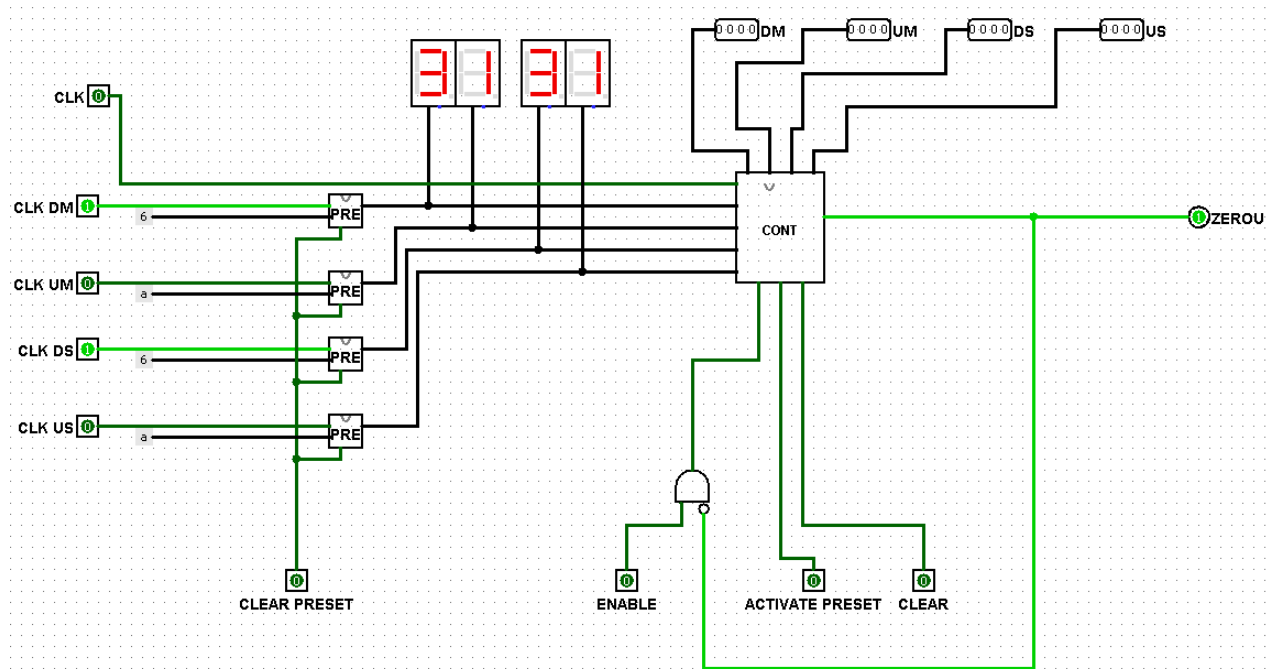


Figura 10: Cronômetro

Com o contador regressivo pronto basta construir a lógica de funcionamento do cronômetro.

Utilizando o circuito de preset podemos definir o valor máximo para o algarismo das dezenas (6) e das unidades (10), visto que estamos trabalhando com um marcador de minutos e segundos. Dado isto, o tempo máximo que pode ser escolhido é *59m59s*.

A cada borda de subida dos clocks de preset é aumentado uma unidade do respectivo input de tempo no relógio. Após setar o tempo de basta apertar o botão **Activate Preset** para salvar o tempo total de jogo no cronômetro. Há também um botão de **Clear Preset**, que zera todos os valores de entrada no **preset**.

Para parar o relógio quando o tempo acaba foi conectado a saída **zerou** do contador regressivo a uma entrada negada de uma porta **and**, conectada ao **enable** do contador. Desta forma, quando o cronômetro chega a zero, a passagem de energia é inibida, e o relógio para de funcionar, até serem alterados os valores a partir do **preset**.

2.8 Relógios

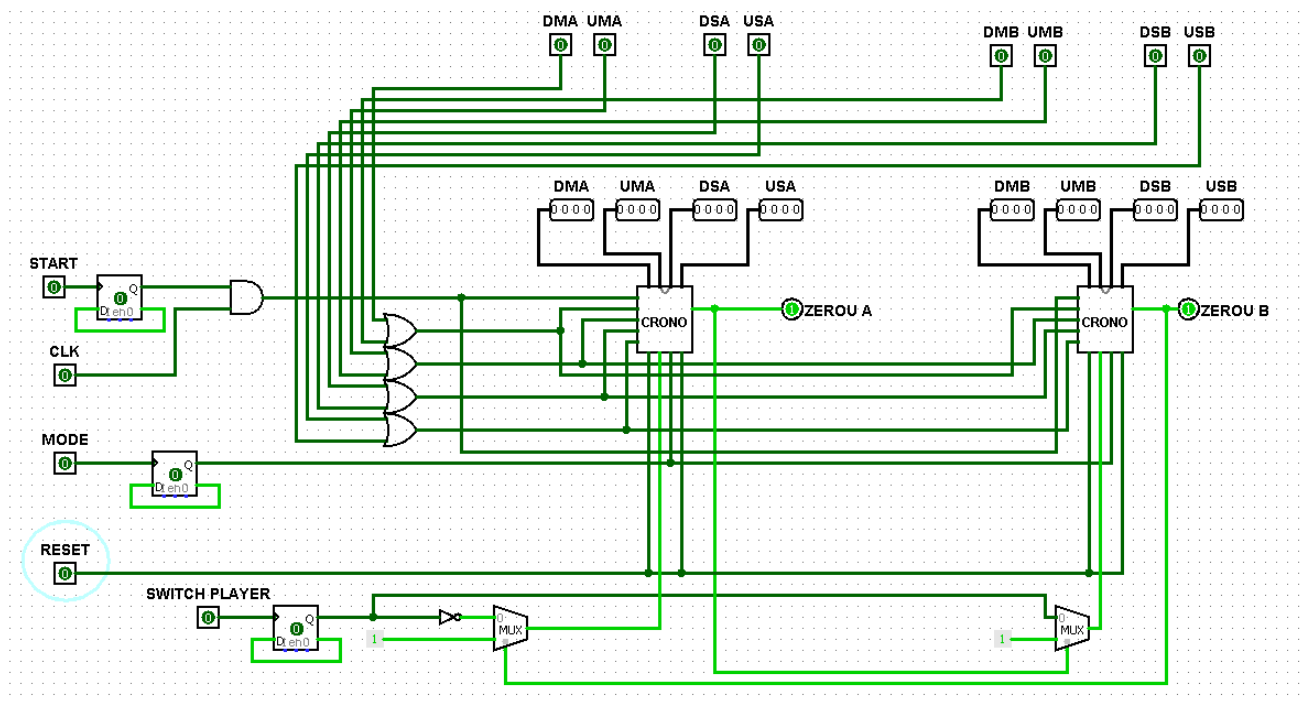


Figura 11: Relógios

O passo final para a contagem de tempo é unir dois cronômetros, para marcar a passagem de tempo dos dois jogadores.

A grande diferença entre o cronômetro visto anteriormente e esses é a conexão entre os dois.

- Caso um dos cronômetros zere o outro estará sempre ativado. Para isto, basta utilizar um multiplexador que passe 1 sempre que o outro cronômetro estiver zerado.
- O botão de **reset** está conectado tanto ao clear do circuito de preset, quanto ao clear do cronômetro.
- O botão de **Switch Player** troca qual o cronômetro estará funcionando. Para isto, o **enable** do cronômetro *A* está conectada negada ao switch, e o do *B* conectada direto. Quando o nível do **Switch Player** for baixo o cronômetro de *A* funcionará, caso contrário o de *B*.

O botão **Mode** ativa o modo de configuração dos relógios, permitindo alterar o tempo de cada um. As entradas de clock dos relógios são conectados por portas *OR*, podendo o tempo ser alterado por qualquer um dos jogadores, ou seja, o tempo configurado para um será o mesmo para o outro.

Como no circuito principal do jogo utilizaremos botões foi decidido utilizar flip-flops para guardar os valores. Caso seja apertado uma vez o botão é ativado, ao apertar de novo é desligado. Para isso, a saída Q' do flip flop é conectada à entrada **D** deste.

2.9 Negativo

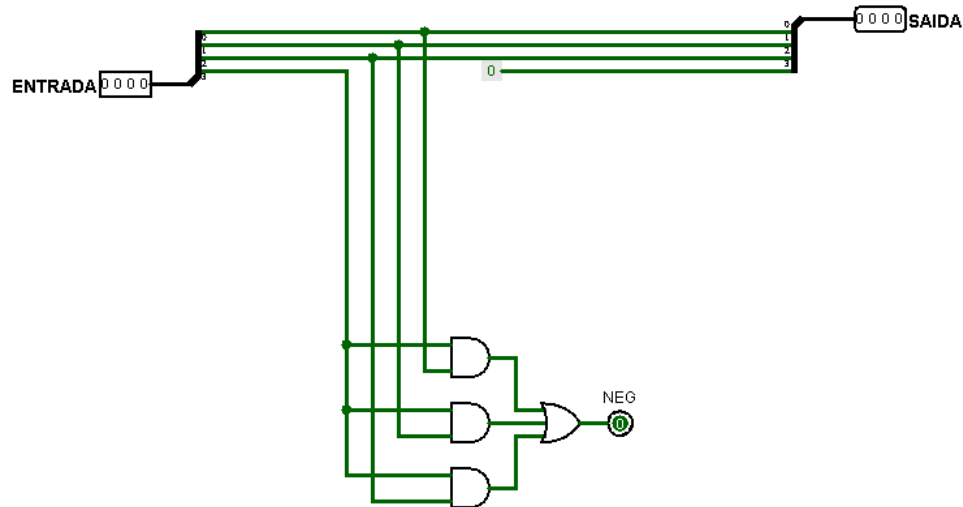


Figura 12: circuito negativo

O circuito negativo recebe um número de 4 bits e o retorna normalmente, desde que o bit de maior significância seja zero. Entretanto, se o bit de maior significância for 1, juntamente com os outros bits, a saída 'neg' será 1, indicando, assim, que o número é negativo. Ao receber, por exemplo, valores como '1011', esse circuito identificará um valor negativo e o converterá em '0011', transformando os valores que excedem '1000' em valores negativos.

2.10 Circuito Somador

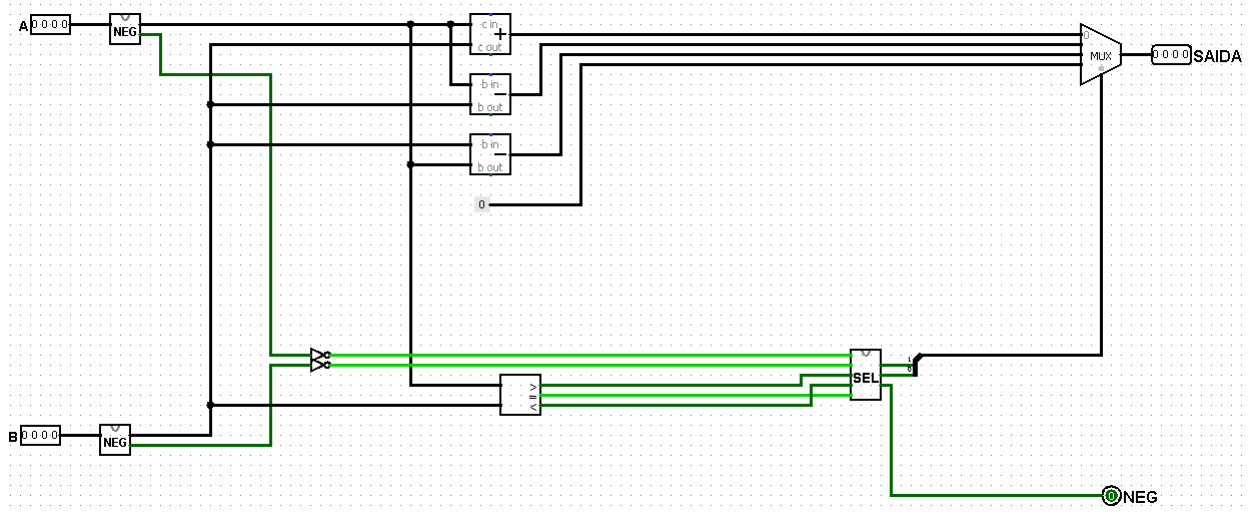


Figura 13: Circuito Somador

O circuito somador soma os dois números de entrada (A e B) considerando seus sinais.

Na entrada do circuito ambos os números passam pelo circuito **negativo**, gerando valores com módulo menor que 7 com seu sinal. É então necessário considerar alguns casos:

OBS: Considere que neg indica se a soma é negativa ($neg = 1$) ou positiva ($neg = 0$).

1. $A \geq 0$ e $B \geq 0$: retorna $A + B$ e $neg = 0$
2. $A < 0$ e $B < 0$: retorna $|A + B|$ e $neg = 1$
3. $A > 0$ e $B < 0$:
 - se $|A| > |B|$: retorna $|A| - |B|$ e $neg = 0$
 - se $|A| < |B|$: retorna $|B| - |A|$ e $neg = 1$
 - se $|A| = |B|$: retorna 0 e $neg = 0$
4. $A < 0$ e $B > 0$:
 - se $|A| > |B|$: retorna $|A| - |B|$ e $neg = 1$
 - se $|A| < |B|$: retorna $|B| - |A|$ e $neg = 0$
 - se $|A| = |B|$: retorna 0 e $neg = 0$

É utilizado um circuito seletor para selecionar as entradas do seletor do multiplexador considerando os casos acima apresentados.

2.11 Comparador

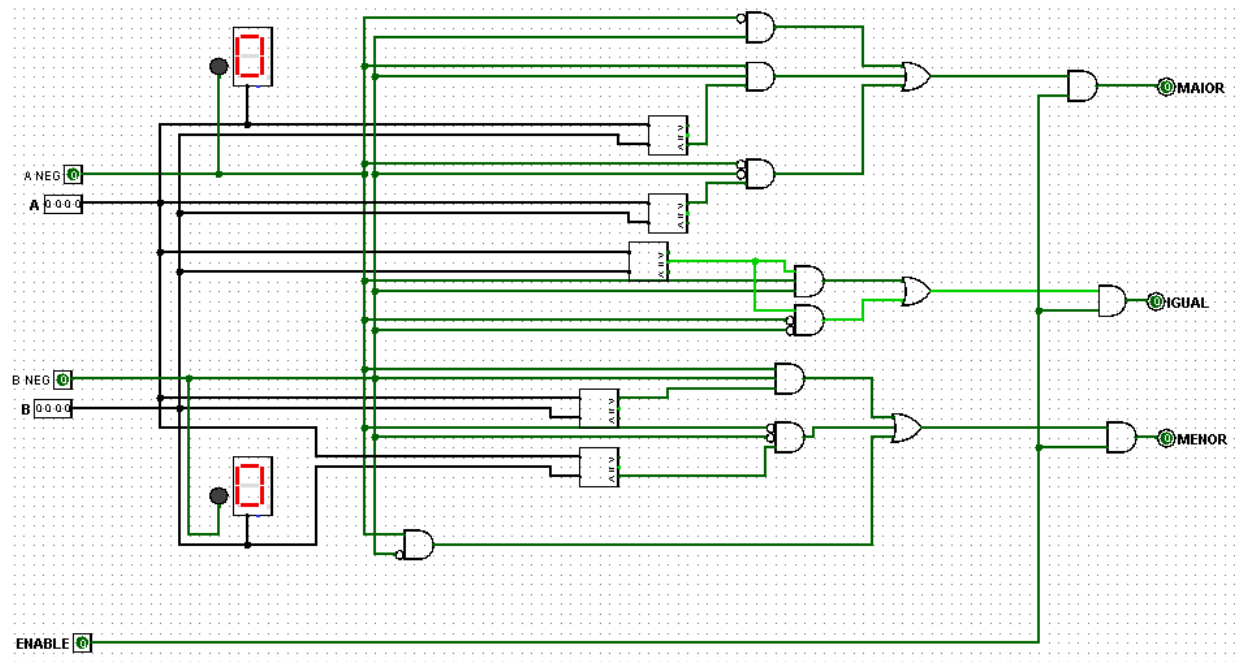


Figura 14: circuito comparador

O circuito comparador, a princípio, recebe dois números de 4 bits para compará-los. Os números recebidos podem ser negativos ou não, e, nesse caso, a entrada é acompanhada pela entrada auxiliar A NEG ou B NEG. Dessa forma, o circuito prossegue para realizar a comparação de magnitude, onde o módulo do valor é comparado com o da outra entrada. Simultaneamente, o circuito também compara os sinais dos valores. Se um número tem um módulo maior e não é negativo, o circuito retorna que ele é maior, e de forma análoga para outros casos.

2.12 Circuito Próximo

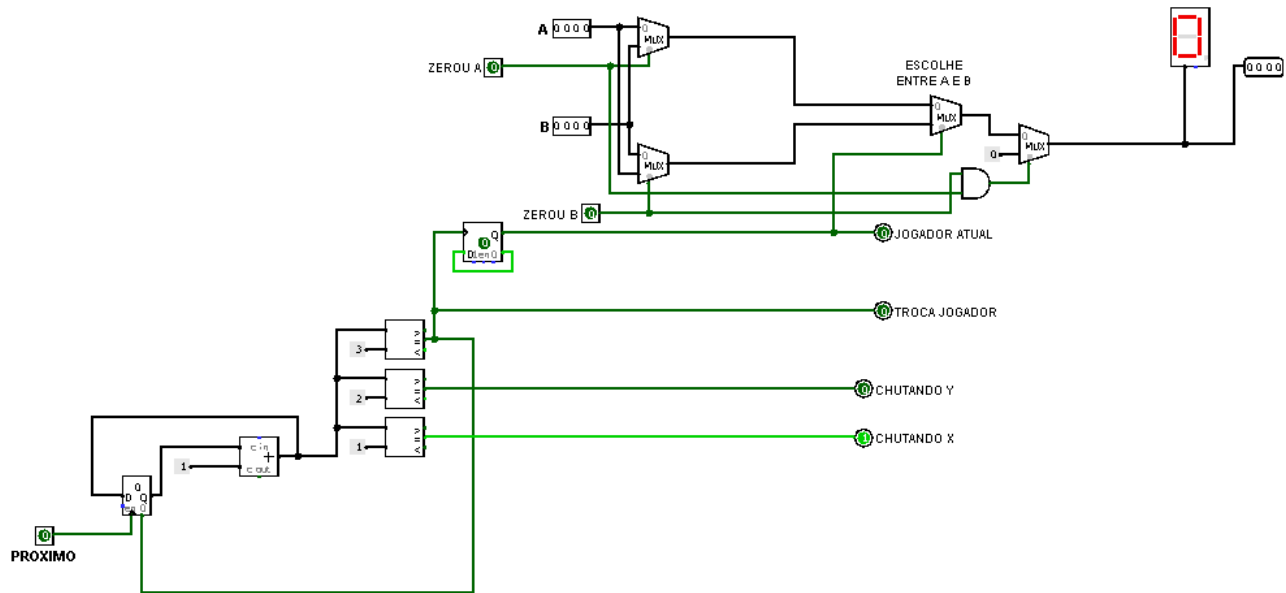


Figura 15: circuito proximo

No circuito, Proximo, ao início de cada acionamento da entrada, o somador entra em ação e incrementa o valor em 1. Imediatamente, um comparador avalia o valor, e se for igual a 1, indica que o chute do jogador X está ativo, e que Após o próximo acionamento, o valor 2 é obtido, e mais uma vez, um dos comparadores verifica se o valor é igual a 2, retornando que o chute é do jogador Y.

Da mesma forma, outro comparador sinaliza a troca de jogadores e inicia novamente o ciclo, passando o chute do jogador atual. Esse chute do jogador em atividade passa por diversos multiplexadores relacionados ao cronômetro de cada jogador, assegurando que a jogada do respectivo jogador só ocorra se ele ainda tiver tempo disponível. Dessa maneira, o circuito permite apenas jogadas válidas, e caso um jogador esgote seu tempo, todas as jogadas subsequentes serão feitas pelo outro jogador, até que eventualmente esgote seu tempo também.

2.13 Core

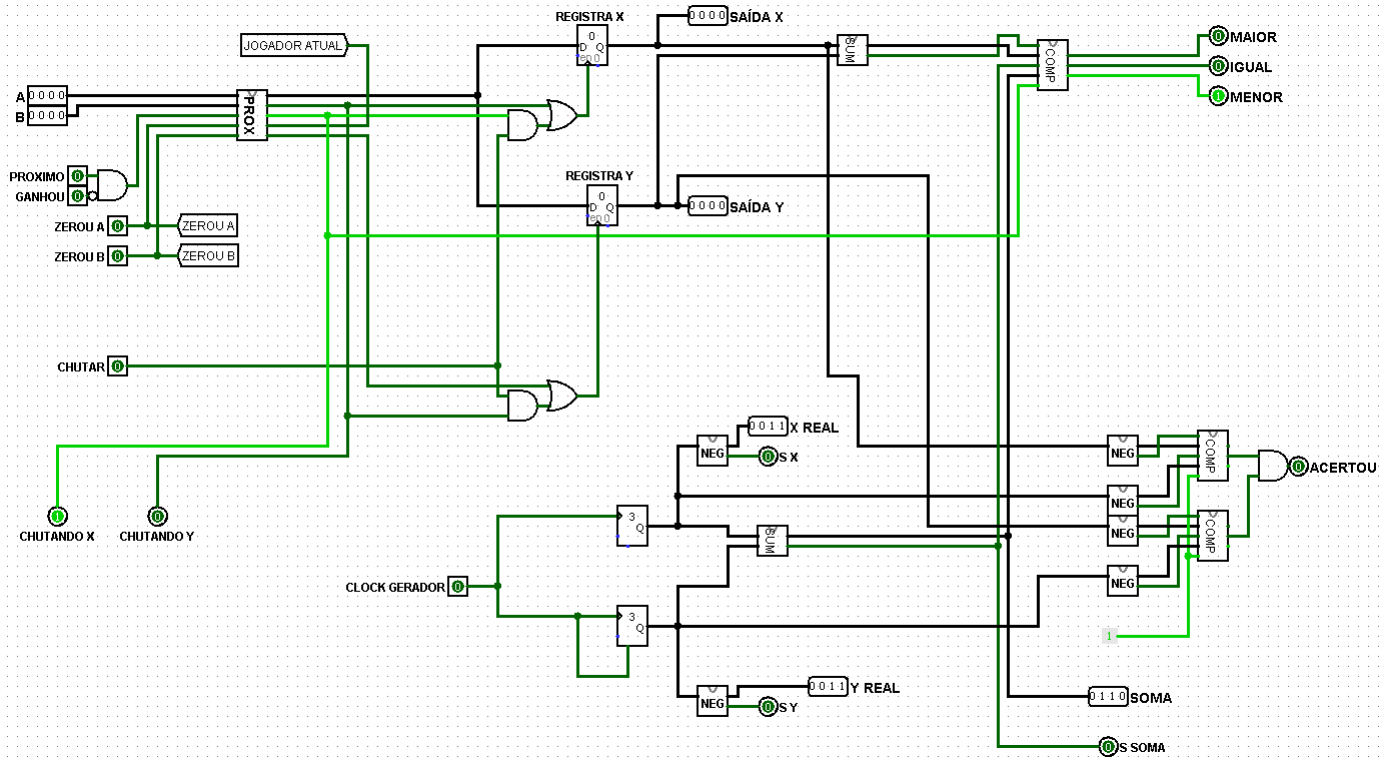


Figura 16: Circuito Core

O circuito **Core** é o circuito principal do jogo. Ele é responsável por armazenar os chutes dos jogadores, alterar as rodadas, dizer se os chutes estão corretos, ou a magnitude das comparações.

Inicialmente, os chutes dos jogadores são coletados pelo circuito **PROX**, que decide a rodada atual, qual o chute, e qual o valor que está sendo registrado agora (X ou Y). O clock para o circuito Próximo é um botão próximo ligado em uma porta AND a um botão de ganhou negado, ou seja, o circuito só funcionará até um dos jogadores ganhar, depois disso o circuito não permitirá mais serem passados chutes. O botão **CHUTAR** passa o chute para os registradores sem passar a rodada, apenas é mostrado no display o chute dado. O chute só é computado na troca de jogador.

O circuito **PROX** também mostra quem está jogando atualmente, e qual o chute atual. Isso funcionará para informar aos jogadores aonde estão situados no jogo (quem está jogando, e qual o chute atual).

As entradas de **ZEROU A** e **ZEROU B** servem para indicar se um dos jogadores, ou ambos, perdeu, permitindo apenas ao outro jogador palpitar.

Na parte de baixo do circuito é possível ver um gerador de números aleatórios que gera duas saídas, X e Y , que serão somadas, assim como os chutes do jogador. Os valores da soma são comparados, e informado ao jogador se a soma é maior, menor ou igual à soma dos valores gerados aleatoriamente.

Há também saídas que indicam os valores dos números gerados e da soma destes. Essas saídas apenas servem para auxiliar os programadores a testar se o jogo está funcionando, elas não serão vistas no jogo principal.

Ademais, há uma saída para indicar que o jogador acertou o chute, os valores X e Y simultaneamente, não apenas a soma.

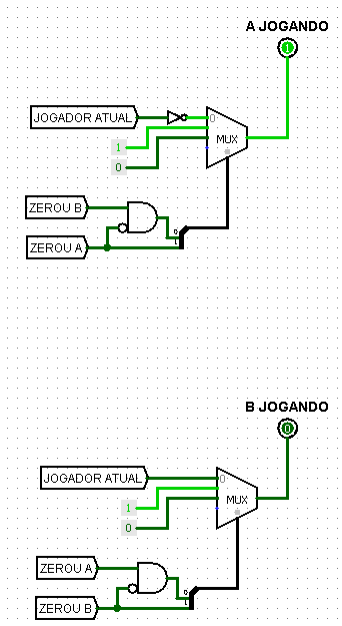


Figura 17: Jogador Atual

Para indicar o jogador atual é conectado a saída **Jogador Atual** a um multiplexador. Caso o tempo do jogador esteja zerado é passado 0 à saída diretamente, caso o outro jogador tenha acabado seu tempo, e o jogador atual ainda tenha tempo é passado 1 diretamente.

2.14 Placar

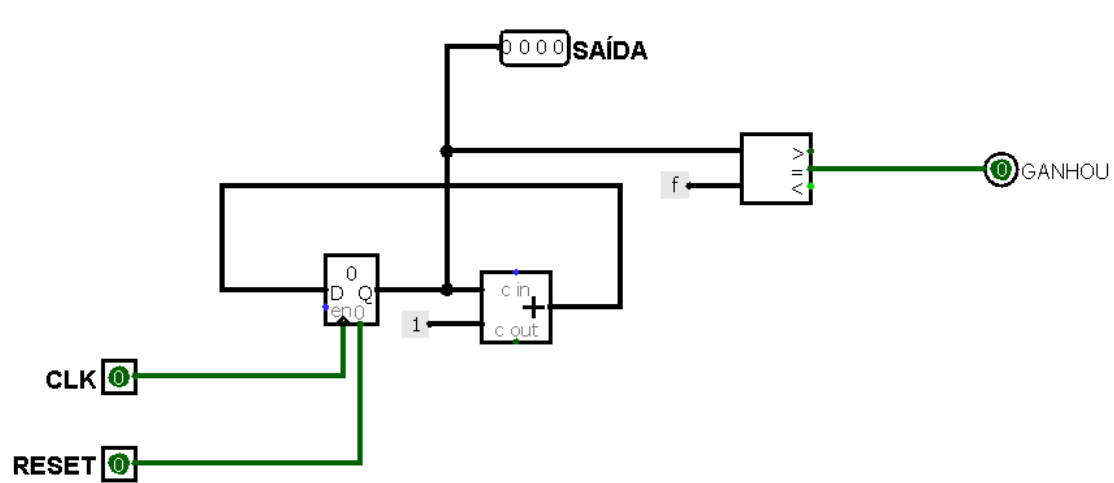


Figura 18: circuito placar

Finalmente, no circuito do placar, após um jogador conseguir acertar o número, o circuito do placar recebe um pulso de clock que adiciona 1 ao valor. Cada jogador tem seu próprio circuito de placar, e ao completar 15 pulsos de clock, o valor "f" é comparado no circuito, indicando a vitória do jogador.

3 Conexões Adicionais no Jogo Principal

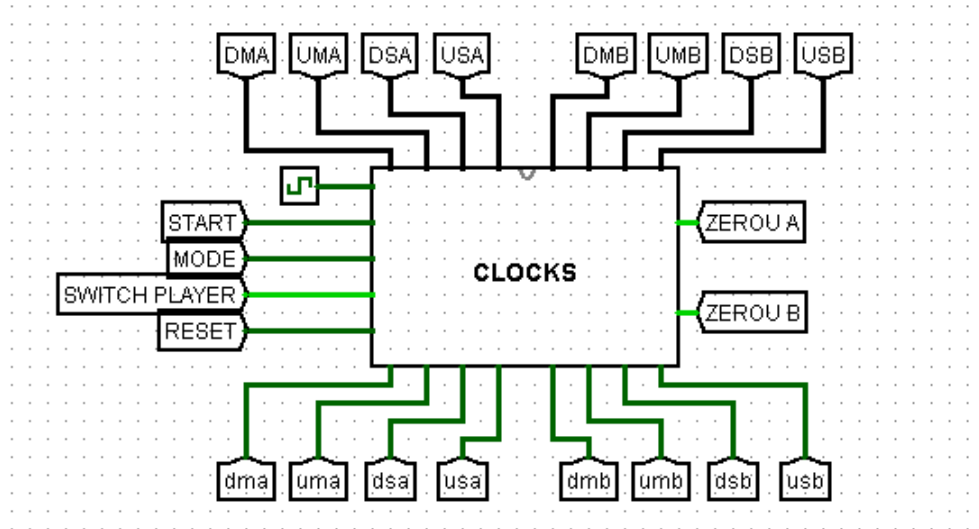


Figura 19: Conexões nos relógios

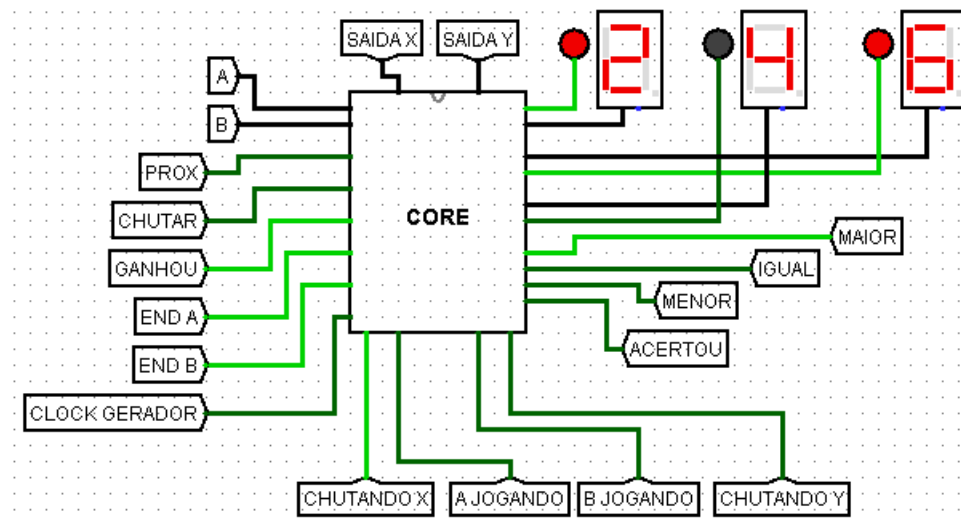


Figura 20: Conexões no circuito Core

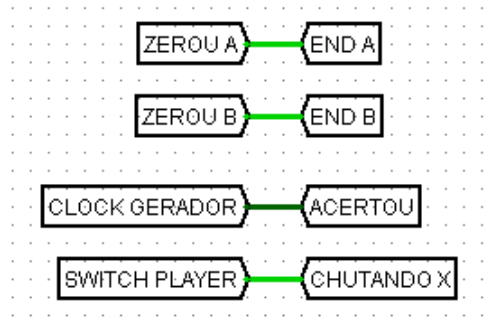


Figura 21: Conexões entre o core e os relógios

As saídas **Zerou A** e **Zerou B** entram no circuito de **CORE** para indicar quando o tempo de algum dos jogadores acabou.

O clock do gerador é ativado quando alguém acertou, indicando que é necessário gerar um novo número aleatório.

Por fim, Sempre que o **Chutando X** ativar significa que a rodada passou, sendo necessário ativar o **Switch Player** no relógio.

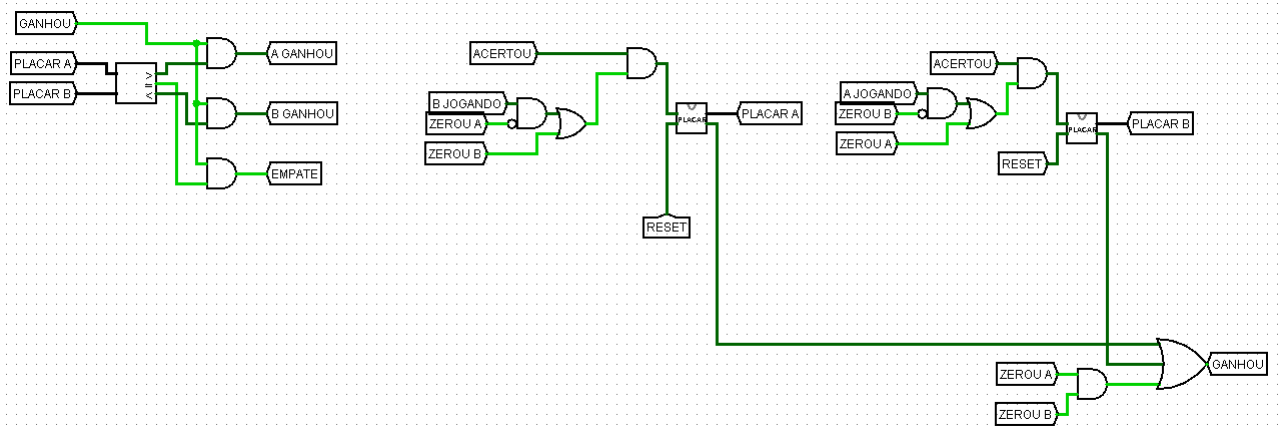


Figura 22: Conexões no placar

Se alguém ganhou é apresentado o resultado (empate, A ganhou ou B ganhou). São também testados os casos dos resultados para analisar se alguém ganhou ou se o jogo encerrou.