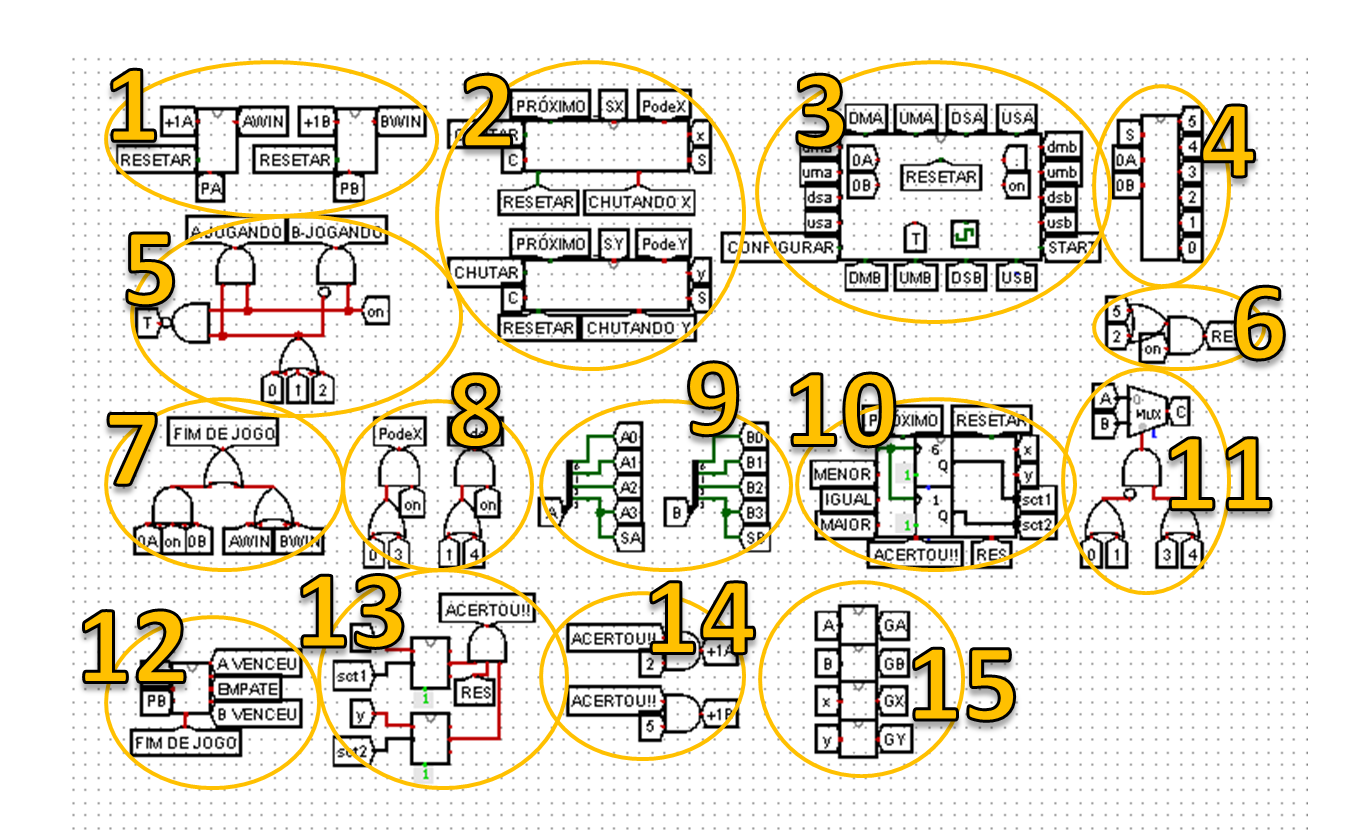
**Relatório de Funcionamento do Circuito  
Guess The Number Extreme**

*Matheus Bezerra Luna*

Projeto proposto pelo professor Ramon Nepomuceno para a disciplina de Circuitos Digitais do curso de Ciência da Computação da Universidade Federal do Cariri (UFCA).

Segue descrição de cada circuito criado no LogiSim para execução do projeto (omite-se a descrição de circuitos básicos da disciplina estudada):

* **Memória:** Representa o Flip-Flop tipo D, com acréscimo da entrada “CHANGE” que tem funcionamento idêntico ao do CLK, mas funciona somente quando o ENABLE está desativado.
* **Cronômetro/1:**   
   Usamos 4 memórias para guardar os 4 bits de cada dígito do cronômetro, sendo o número formado por elas levado a um somador de 4 bits e suas entradas sendo o resultado dessa soma.  
   O segundo número da soma será determinado posteriormente, pois precisamos fazer seu controle para que seja f (equivalente a -1) na maior parte do tempo, mas que seja um número N desejado quando o número do display for 0, para que ao invés de ir de f a 0 o display vá apenas de N a 0 (nesse caso, N será 5 para as dezenas de minuto e segundos e 9 para as unidades, a fim de representar os valores máximos de 59 minutos e 59 segundos).  
   No mais, temos as entradas CLK, CHANGE, ENABLE e CLEAR, que são intuitivas, e as saídas Display, que equivale ao número a ser mostrado no display, e Padrão, que será 1 quando o número do display for 0.
* **Cronômetro/2:**   
   Usamos 4 circuitos Cronômetro/1, um para cada display. Entradas CLK, ENABLE e CLEAR intuitivas. Entradas US, DS, UM e DM são as de configuração dos dígitos do cronômetro, ligadas aos CHANGEs.  
   Ligamos a saída Padrão de cada componente à entrada seletora de um multiplexador que tem como entrada associada ao 0 o valor f e entrada associada a 1 o valor 5 para dezenas e 9 para unidades. A saída do multiplexador é ligada à entrada N\_SOMA de cada componente.  
   As portas AND marcadas com “%” identificam quando o respectivo display atinge 0, As portas AND não marcadas identificam quando todos os dígitos anteriores ao correspondente zeram e as portas AND marcadas com “&” servem para habilitar o ENABLE do componente correspondente somente quando o ENABLE geral está ativado e todos os dígitos anteriores estão zerados. Esse sistema faz com que os dígitos no cronômetro somente diminuam quando os anteriores a ele zeram.  
   Por fim, as saídas D\_min, U\_min, D\_seg e U\_seg são os valores de cada display e a saída Zerado indica que o tempo do jogador terminou (todos os dígitos zerados).
* **Cronômetro/3:**   
   As entradas CONFIG e INICIAR funcionam de forma complementar. Ao apertar RESETAR, a saída da memória de INICIAR será zerada por ativação de seu CLEAR, enquanto que a saída da memória de CONFIG será 1, permitindo a configuração dos cronômetros.   
   Ao pressionar INICIAR, a saída de sua memória será 1 e então o CLEAR da memória de CONFIG será ativado, zerando sua saída. O inverso ocorre ao pressionar CONFIG.  
   A entrada JOGADOR habilita o primeiro cronômetro quando é 0 e o segundo quando é 1, sob hipótese de Jogando estar ativo e o cronômetro correspondente não estar zerado (indicado pelas saídas 0a e 0b).  
   As demais entradas e saídas são intuitivas.
* **Chute:**   
   A parte superior desse circuito é uma máquina de 6 estados. Para que isso seja possível, utilizamos três memórias para representar 3 bits e um somador de 3 bits para incrementar o estado, que recebe como uma das parcelas o números advindo das memórias. A segunda parcela da soma é escolhida no multiplexador, sendo 1 normalmente, e tendo o valor 3 quando o estado 5 é detectado, para que o próximo estado seja 0 e não 6.  
   A parte inferior do circuito tem, além das entradas intuitivas CLEAR e CLK, a entrada X/Y, que receberá o valor da coordenada que o jogador chutou, e a entrada CHUTANDO, que ativa os ENABLEs das memórias, permitindo o armazenamento do valor da coordenada na saída Display.   
   Led será ativado quando o jogador estiver chutando a coordenada correspondente e Sinal é ativado se o número da coordenada for negativo.
* **Situação:**   
   Circuito projetado diretamente a partir de uma tabela verdade, que tem como saídas 6 representações de possíveis situações do jogo, sendo elas, respectivamente:  
   A chutando x, A chutando y, mostrando feedback do chute de A, B chutando x, B chutando y, mostrando feedback do chute de B.  
   A situação corresponde normalmente aos estados enquanto ambos os cronômetros ainda têm tempo. Quando A zera seu cronômetro, a situação passa a percorrer somente 3, 4 e 5, e quando B zera seu cronômetro a situação percorre somente 0, 1 e 2.
* **Jogo/1:**   
   Tem como entradas os valores de X e Y chutados e os valores das duas coordenadas secretas. Utilizando o circuito “4 para 5”, que soma dois números de 4 bits em complemente de 2 e converte o resultado para 5 bits em complemento de 2, somamos X a Y e SCT1 a SCT2, e os resultados comparamos com o circuito “comparador” para indicar se a soma dos números secretos é MENOR, IGUAL ou MAIOR à soma dos números chutados.
* **Jogo/2:**  
   A parte direita do circuito determina o pulso do clock de geração de novos valores aleatórios. Para isso, usamos uma memória em que ao PRESET ligamos a entrada ACERTOU e ao CLEAR ligamos PRÓXIMO, para que quando o jogo indicar que o jogador acertou as coordenadas e PRÓXIMO seja pressionado, novas coordenadas aleatórias sejam geradas (Novas coordenadas também são geradas quando o jogo é resetado).  
   Já a parte direita do circuito é uma implementação de Jogo/1, com o acréscimo de as saídas só indicarem a comparação quando RES está ativado.
* **Comparador:**  
   Comparador de dois números de 5 bits em complemento de 2.  
   A parte superior comparada somente os bits mais significativos (de sinal), e caso eles sejam iguais a parte inferior é um comparador simples entre os demais 4 bits.
* **4 para 5:**  
   Circuito que soma dois números de 4 bits em complemento de 2, mostrando o resultado em um número de 5 bits em complemento de 2.  
   O seguimento superior representa a situação em que A e B possuem sinais distintos (e portanto a soma dos dois não gera overflow). Assim, simplesmente somamos os números com um somador de 4 bits, então tomamos seu módulo e acrescentamos um 0 ao 5° bit (pois o módulo é sempre positivo) para a entrada 0 de um multiplexador, e para a entrada 1 tomamos o número negativo com esse módulo.   
   O bit de seleção será o sinal do número obtido da soma de 4 bits, fazendo com que o módulo seja selecionado no caso de ser um número positivo, e o número negativo caso contrário.  
   O seguimento inferior refere-se à situação de igualdade de sinais das parcelas. Nesse caso, somamos os módulos dos números, gerando um número de 5 bits (aproveitando o cout) que tem bit mais significativo 0, que ligamos à entrada 0 de um multiplexador. A entrada 1 recebe o número negativo com esse módulo. O bit de seleção é o sinal de A (que é igual ao de B), pois se o sinal é negativo, a soma também será, e vice-versa.  
   Por fim, o resultado do seguimento superior vai à entrada 0 de um multiplexador, e o do seguimento inferior para a entrada 1. O bit de seleção é resultado de um XNOR entre os sinais de A e B, exatamente porque se os sinais são distintos a saída do XNOR será 0, e se forem iguais será 1.
* **Placar:**  
   Contador simples, a menos da saída Venceu que detecta quando o número do display atingiu 15 (pontuação máxima).
* **Comparador simples.**
* **Módulo:**  
   Retorna o valor absoluto de um número de 4 bits em complemento de 2.  
   Na entrada 0 de um multiplexador levamos o próprio número, e na entrada 1 o complemento de 2 desse número. O bit de seleção é o sinal do número. Assim, se ele for positivo, ele próprio irá para a saída e, se for negativo, seu complemento que irá.
* **Soma Simples.**
* **Somador 3 bits.**
* **Somador 4 bits.**
* **Somador 5 bits.**
* **Main:** O funcionamento do jogo está descrito no pdf da proposta do projeto feito pelo professor Ramon.   
   Sugestão: Ao abrir o arquivo do circuito no LogiSim, aperte CTRL+K para ativar o clock e na aba Simular mude a frequência de pulso para 2Hz, a fim de obter um funcionamento dos cronômetros similar ao real.   
   Segue abaixo a lógica de cada componente usado no main:



1. Implementação dos placares. +1A indica quando se deve aumentar a pontuação de A, RESETAR limpa o placar e PA indica o valor a ser mostrado no display. Análogo para B.  
 2. Implementação da lógica de chutes. C indica qual jogador está chutando (A ou B), x (y) indica o valor chutado, S indica o estado do jogo, PodeX (Pode Y) indica se a situação atual é de coletar x (y), SX (SY) indica o sinal do valor chutado e CHUTANDO X (CHUTANDO Y) liga o respectivo led quando se está chutando a coordenada.  
 3. Implementação dos cronômetros. DMA, UMA, DSA, USA (DMB, UMB, DSB, USB) identificam ativações dos botões de configurar os cronômetros, as mesmas tags em minúsculo indicam os valores para cada display do cronômetro, 0A e 0B determinam se os cronômetros estão zerados, “.” Indica se o jogo está em modo de configuração, “on” se jogo está rodando, T indica qual jogador está em sua vez, START e CONFIGURAR identificam o acionamento dos respectivos botões e o CLOCK é o clock geral de todo o jogo.  
 4. Implementação da situação de jogo. S é o estado atual, 0A e 0B indicam se algum cronômetro está zerado, e 0, 1, 2, 3, 4, 5 são as possíveis situações.  
 5. Determina qual jogador está em sua vez (T), e determina a situação dos leds A-JOGANDO e B-JOGANDO. No caso, a vez será de A (T=0) quando o jogo estiver rodando (on) e a situação for 0, 1 ou 2. Do contrário, B esta jogando.  
 6. Determina os momentos em que se deve dar feedback ao jogador, que são quando o jogo está rodando e a situação é 2 ou 5.  
 7. Determina quando o jogo termina. Isso ocorre quando o jogo está rodando e ambos os cronômetros zeraram, ou quando A ou B ganham (AWIN ou BWIN).  
 8. Determina os momentos em que se está recebendo o valor ou x valor de y. Que são quando o jogo está ativo e a situação é 0 ou 3 para x e 1 ou 4 para y.  
 9. Determina os valores de A e B e de seus sinais a partir dos bits selecionados.  
 10. Implementação da lógica do jogo. As entradas x e y são os valores chutados, sct1 e sct2 são os números secretos, que serão determinados por geradores de valores aleatórios, e o clock determinado dentro do circuito Jogo/2 é ligado ao clock dos geradores. RES indica se a situação é de feedback, ACERTOU!! indica se os chutes são correspondentes aos valores secretos, e MENOR, IGUAL e MAIOR indicam a comparação entre as somas dos valores chutados e dos valores secretos.  
 11. Determina o valor C que será usado no chute, que corresponderá ao número do jogador da vez. Assim, se a situação é 0 ou 1 passamos o valor de A, e se for 3 ou 4 passamos o valor de B.  
 12. Implementação do feedback de final de jogo. Aqui, utilizamos um comparador simples entre os valores dos placares (PA e PB), caso FIM DE JOGO esteja ativado, indicando que o jogo terminou. Obtemos então uma saída para A VENCEU, EMPATE ou B VENCEU, dependendo da comparação.  
 13. Determinação de acertou das coordenadas. Usamos dois comparadores simples para comparar se x = sct1 e se y = sct2. Caso isso ocorra e a situação seja de feedback (RES), ativamos ACERTOU!!.  
 14. Lógica de ganho de pontos. Se ACERTOU!! está ativo e a situação é 2 (5), ativamos +1A (+1B) para que a pontuação do jogador aumente.   
 15. Converte os valores dos números em complemento de 2 que precisarão ser mostrados em displays para seus módulos.