

Descrição da ULA

Carolina Ferreira e Mateus Caçabuena

Como configurar a ULA para utilização:

REGISTRADORES:

Input de valores:

- **Registrador A:** A entrada de valores ocorre por dois caminhos. O usuário pode inserir o valor através do botão “presetA”. Além disso, o resultado do multiplexador é inserido na entrada do primeiro Flip-flop, na entrada D7.
- **Registrador B:** A entrada também pode ser realizada de duas maneiras: ou pelo botão D (inserindo um valor no primeiro Flip-flop e passando-o adiante por meio de pulsos clock), ou pelo botão “presetB”.

Botões preset e clear:

Para realizar o input manual de valores, utilizou-se as entrada “Preset” e “Clear” dos flip-flops, e a utilização deve ser feita da seguinte forma:

- ❖ Inserir os valores desejados no botão “preset” do devido registrador. Feito isso, os valores serão inseridos nos flip-flops e deve-se zerar o “preset”. O valor a ser executado deve estar armazenado dentro do registrador e exibido nos displays corretamente.
- ❖ Se, por acaso, inserir um valor errado, deve-se selecionar o botão “clear” do registrador (deixando-o com valor 1). Isto irá zerar todos os valores do registrador. Após isso, pode-se alterar o valor desejado no “preset”. Por fim, coloca-se o valor do “clear” em 0 novamente e deve-se zerar os valores do “preset” também. O valor a ser executado deve estar armazenado dentro do registrador e exibido nos displays corretamente.

AND:

Passo a passo:

- ❖ Selecionar o seu primeiro valor de 8 bits no “presetA”.
- ❖ Selecionar o seu segundo valor de 8 bits no “presetB”.
- ❖ Todos os pinos selecionados como “1” nos registradores deverão ser selecionados novamente, a fim de todos os 8 números retornarem a 0 e a operação ocorrer corretamente sem haver conflitos.
- ❖ No multiplexador, mantenha “0” no pino “select[1]” e faça o mesmo no pino “select[0]”, isto fará o circuito entender que você deseja realizar a operação AND entre cada bit de ambos registradores.
- ❖ No clock, realize 8 pulsos, ao final do oitavo pulso, você receberá seu novo operando no HEX display do registradorA.

OR:

Passo a passo:

- ❖ Selecionar o seu primeiro valor de 8 bits no “presetA”.
- ❖ Selecionar o seu segundo valor de 8 bits no “presetB”.
- ❖ Todos os pinos selecionados como “1” nos registradores deverão ser selecionados novamente, a fim de todos os 8 números retornarem a 0 e a operação ocorrer corretamente sem haver conflitos.
- ❖ No multiplexador, selecione “0” no pino “select[1]” e selecione “1” no pino “select[0]”, isto fará o circuito entender que você deseja realizar a operação OR entre cada bit de ambos registradores.
- ❖ No clock, realize 8 pulsos, ao final do oitavo pulso, você receberá seu novo operando no HEX display do registradorA.

NOT:

A operação NOT está sendo realizada dentro do Multiplexador (MUX). Como controla-se o MUX por meio dos botões “select”, e cada um possui 2 saídas sendo 1 negada, o funcionamento do MUX está diretamente relacionado à operação NOT. Ao observar a fórmula booleana de um MUX 2X1 entende-se melhor a relação entre NOT e o MUX: $OUT = SEL' \cdot IN1 + SEL \cdot IN2$

Ou seja, o OUT do MUX (seja 2x1 ou 4x1) está diretamente relacionado à negação (NOT) do “select”.

Passo a passo:

- ❖ Se select[1] = 0 e select[0] = 0, a operação realizada será AND (primeira entrada).
- ❖ Se select[1] = 0 e select[0] = 1, a operação realizada será OR (segunda entrada).
- ❖ Se select[1] = 1 e select[0] = 0, a operação realizada será XOR/SUB (terceira entrada).
- ❖ Se select[1] = 1 e select[0] = 1, a operação realizada será ADD (quarta entrada).

XOR:

Passo a passo:

- ❖ Selecionar o seu primeiro valor de 8 bits no “presetA”.
- ❖ Selecionar o seu segundo valor de 8 bits no “presetB”.
- ❖ Todos os pinos selecionados como “1” deverão ser selecionados novamente, a fim de todos os 8 números retornarem a 0 e a operação ocorrer corretamente sem haver conflitos.
- ❖ No multiplexador, selecione “1” no pino “select[1]” e mantenha o pino “select[0]” no “0”, isto fará o circuito entender que você deseja realizar a operação XOR entre cada bit de ambos registradores.
- ❖ No clock, realize 8 pulsos, ao final do oitavo pulso, você receberá seu novo operando no HEX display do registradorA.

ADD (adição):

Passo a passo:

- ❖ Selecionar o seu primeiro valor de 8 bits no “presetA”.
- ❖ Selecionar o seu segundo valor de 8 bits no “presetB”.
- ❖ Todos os pinos selecionados como “1” deverão ser selecionados novamente, a fim de todos os 8 números retornarem a 0 e a soma ocorrer corretamente sem haver conflitos.
- ❖ No multiplexador, selecione “1” no pino “select[1]” e faça o mesmo no pino “select[0]”, isto fará o circuito entender que você deseja realizar uma soma.
- ❖ No clock, realize 8 pulsos, ao final do oitavo pulso, você receberá seu novo operando no HEX display do registradorA.

SUB (subtração):

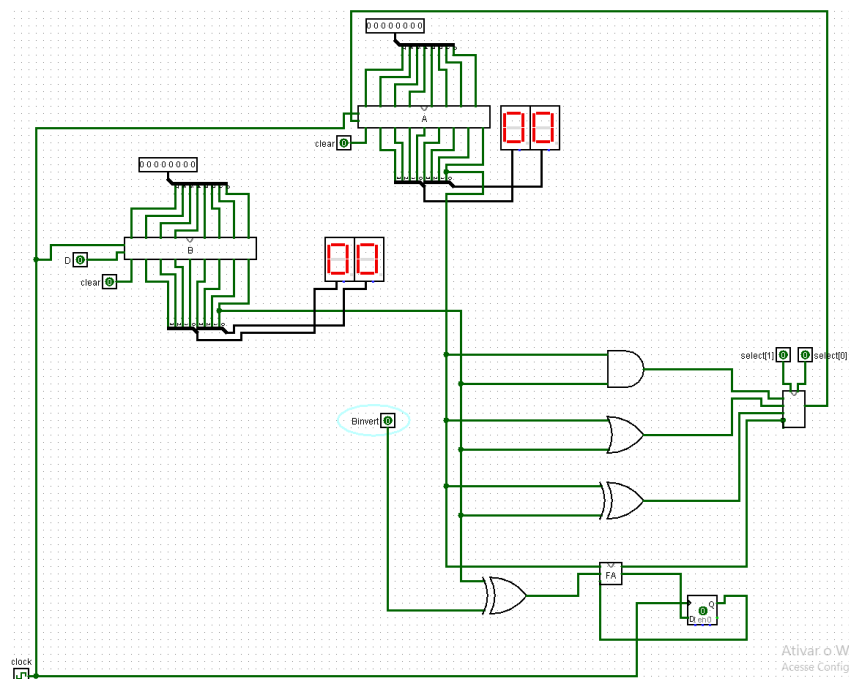
Passo a passo:

- ❖ Selecionar o seu primeiro valor de 8 bits no “presetA”.
- ❖ Selecionar o seu segundo valor de 8 bits no “presetB”.
- ❖ o multiplexador, selecione “1” no pino “select[1]” e faça o mesmo no pino “select[0]”.
- ❖ Por fim, selecione “1” no Binvert e no Carry, isto é o que diferenciara a operação de uma adição.
- ❖ No clock, realize 8 pulsos, ao final do oitavo pulso, você receberá seu novo operando no HEX display do registradorA.

EXEMPLOS DE USO:

Os exemplos serão mostrados através de imagens capturadas do momento em que foram executadas as operações.

Imagem ilustrativa do circuito pronto para ser utilizado:



AND

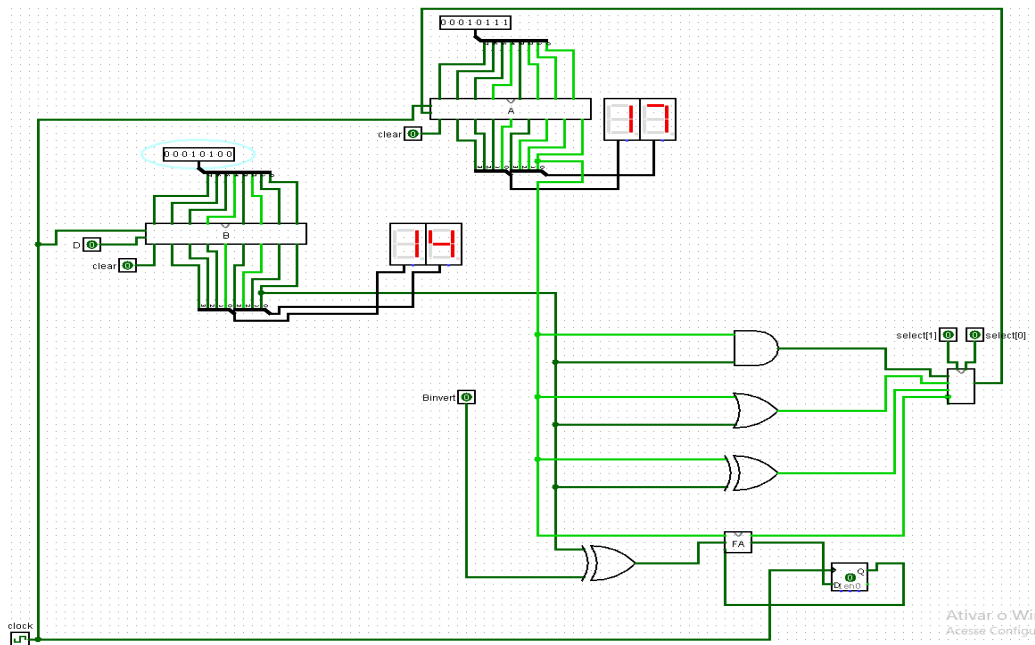
Ex.: 1

RegistradorA = 00010111 (17)

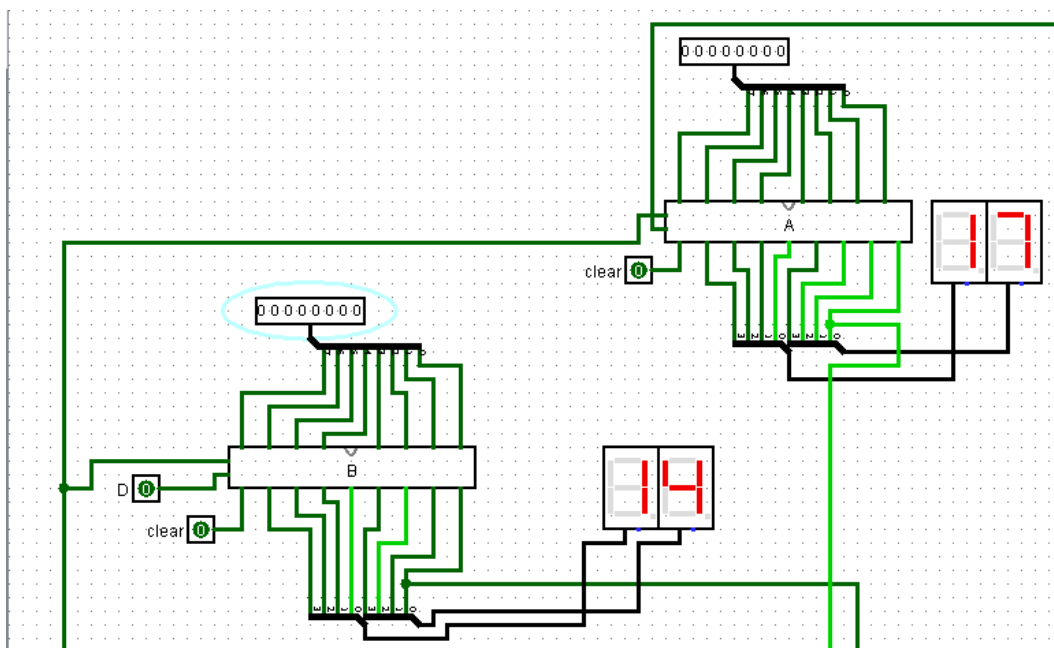
RegistradorB = 00010100 (14)

Resultado esperado = 00010100 (14)

Seguindo o passo a passo da operação AND, como foi escrito acima, a primeira ação é selecionar os valores desejados nos registradores, após isso, selecionar "0" tanto no select[1], quanto no select[0] do multiplexador.



Detalhe importante: é necessário zerar todos os números alterados no registrador para não haver conflito na operação:



Note que os valores hexadecimais não saíram, pois já foram computados e estão prontos para a execução da operação.

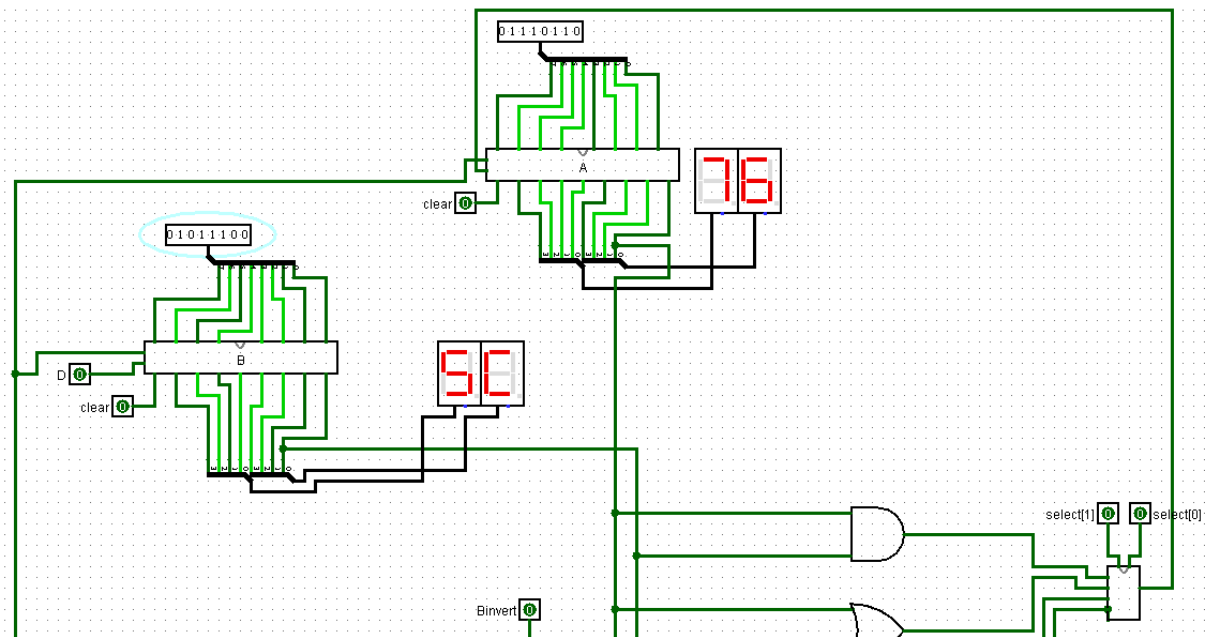
Ex.: 2

RegistradorA = 01110110 (76)

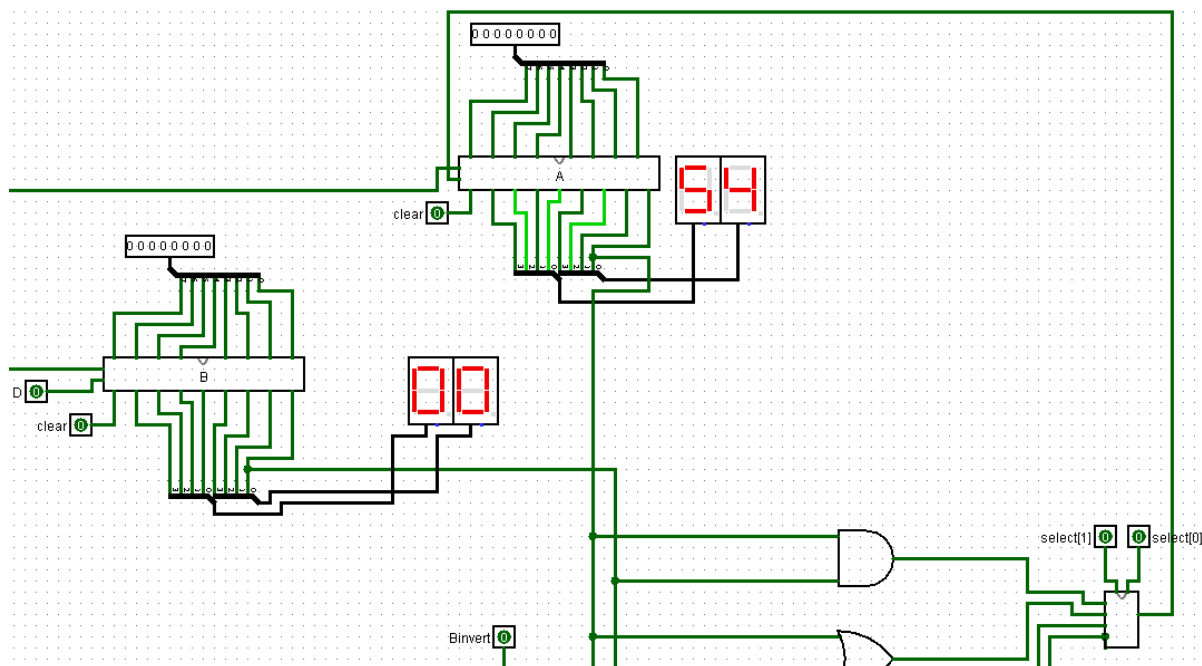
RegistradorB = 01011100 (5C)

Resultado esperado = 01010100 (54)

Seguindo o passo a passo da operação AND, como foi escrito acima, a primeira ação é selecionar os valores desejados nos registradores, após isso, selecionar “0” tanto no select[1], quanto no select[0] do multiplexador.



Após zerar os números alterados e realizar os 8 pulsos do clock, o resultado é ilustrado no registradorA: 01010100 (54)



OR

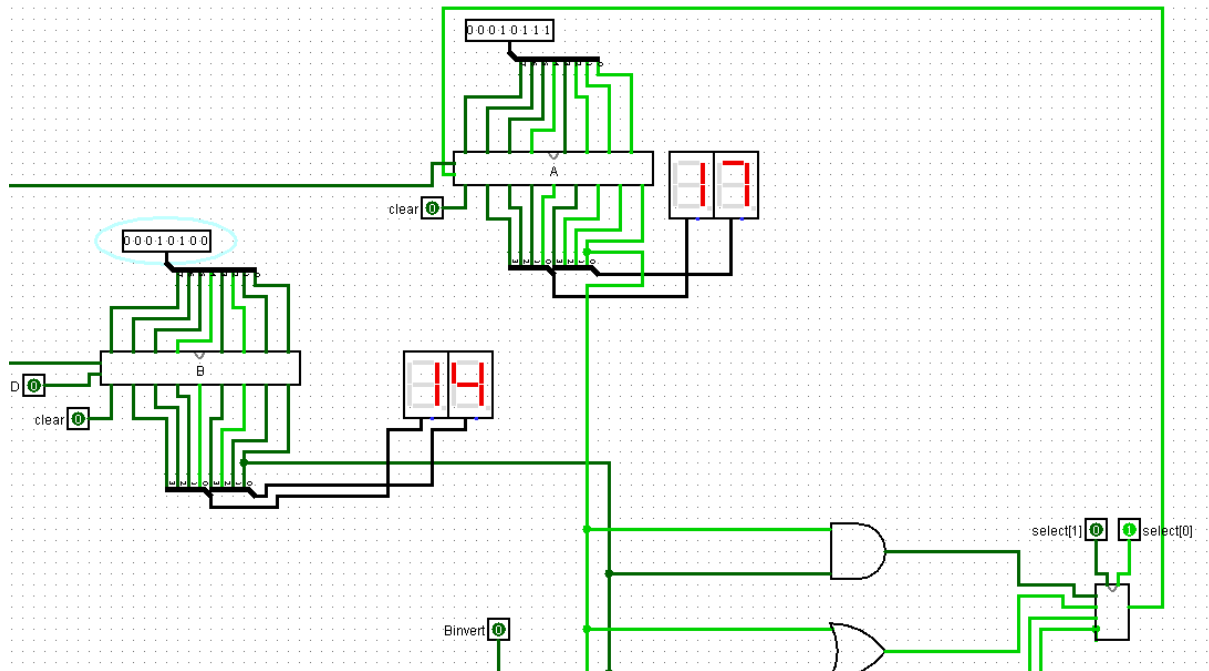
Ex.: 1

RegistradorA = 00010111 (17)

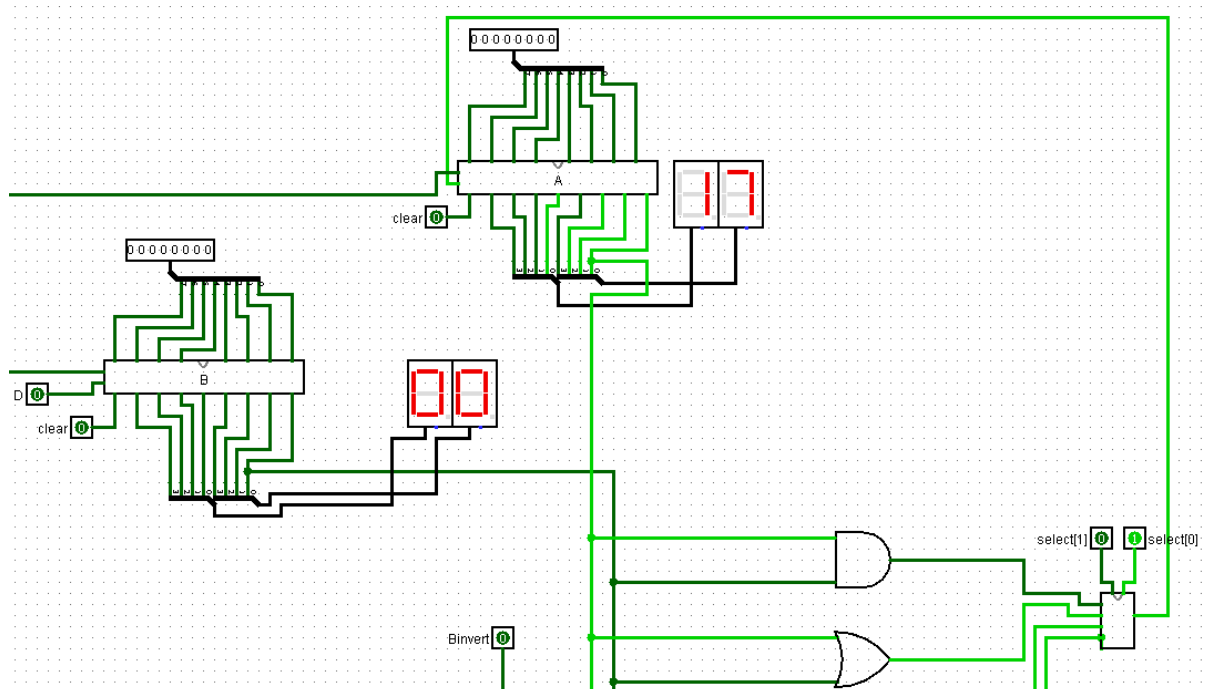
RegistradorB = 00010100 (14)

Resultado esperado = 00010111 (17)

Seguindo o passo a passo da operação OR, como foi escrito acima, a primeira ação é selecionar os valores desejados nos registradores, após isso, selecionar "0" no select[1] e "1" no select[0] do multiplexador.



Após zerar os números alterados e realizar os 8 pulsos do clock, o resultado é ilustrado no registradorA: 00010111 (17)



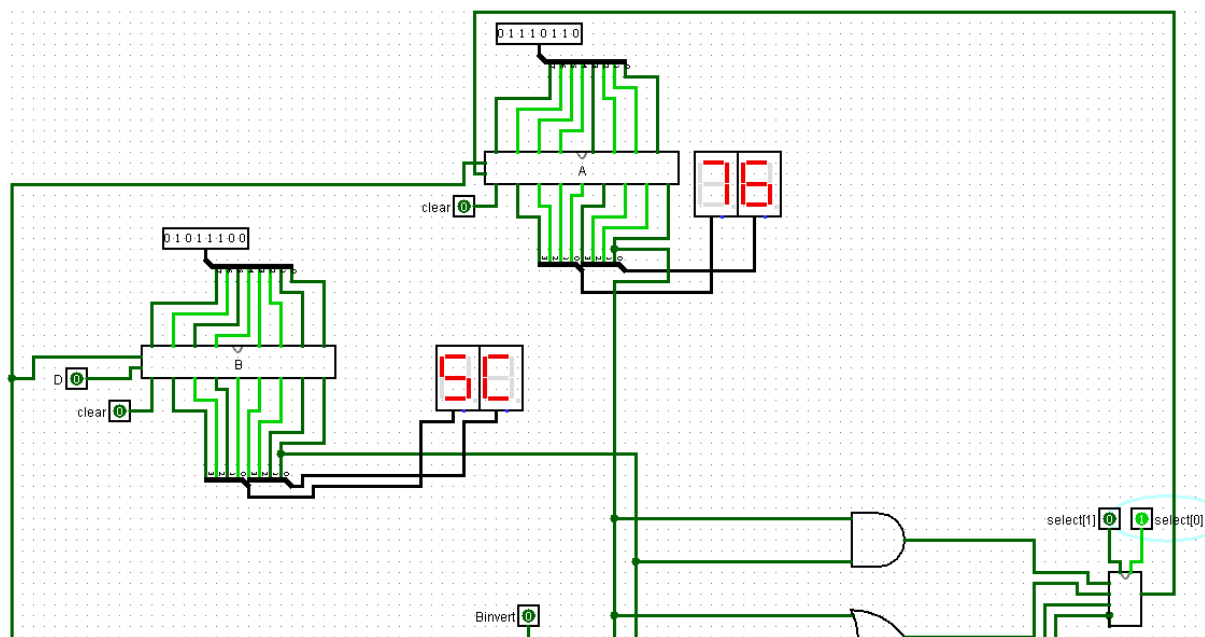
Ex.: 2

RegistradorA = 01110110 (76)

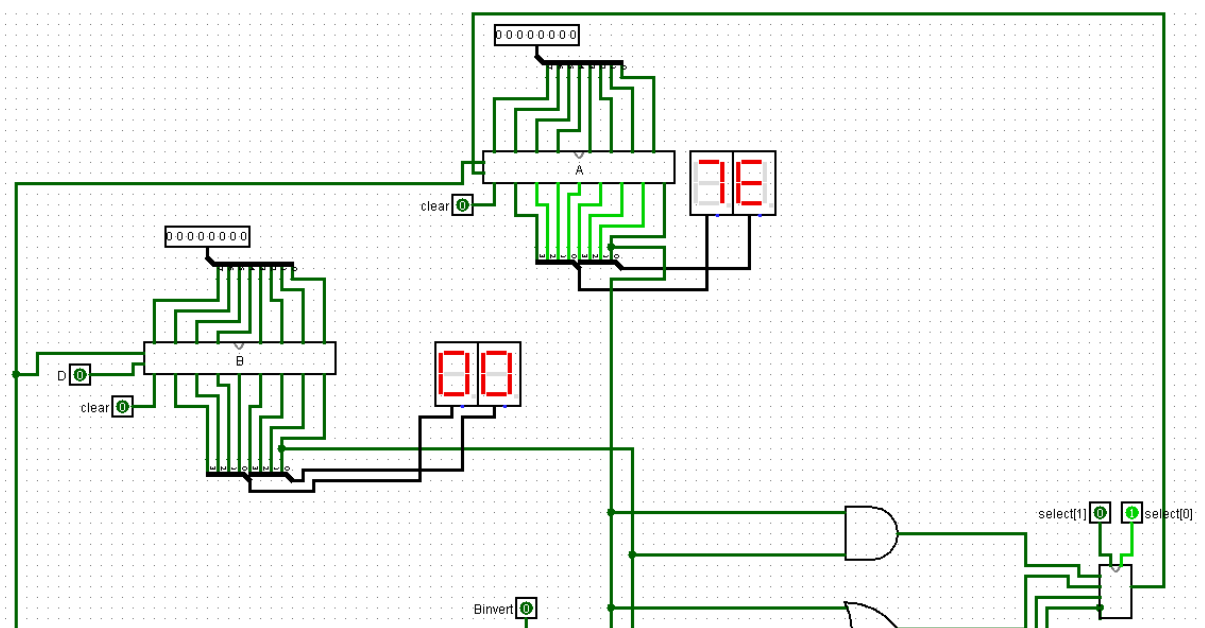
RegistradorB = 01011100 (5C)

Resultado esperado = 01111110 (7E)

Seguindo o passo a passo da operação OR, como foi escrito acima, a primeira ação é selecionar os valores desejados nos registradores, após isso, selecionar “0” no select[1] e “1” no select[0] do multiplexador.



Após zerar os números alterados e realizar os 8 pulsos do clock, o resultado é ilustrado no registradorA: 01111110 (7E)



NOT

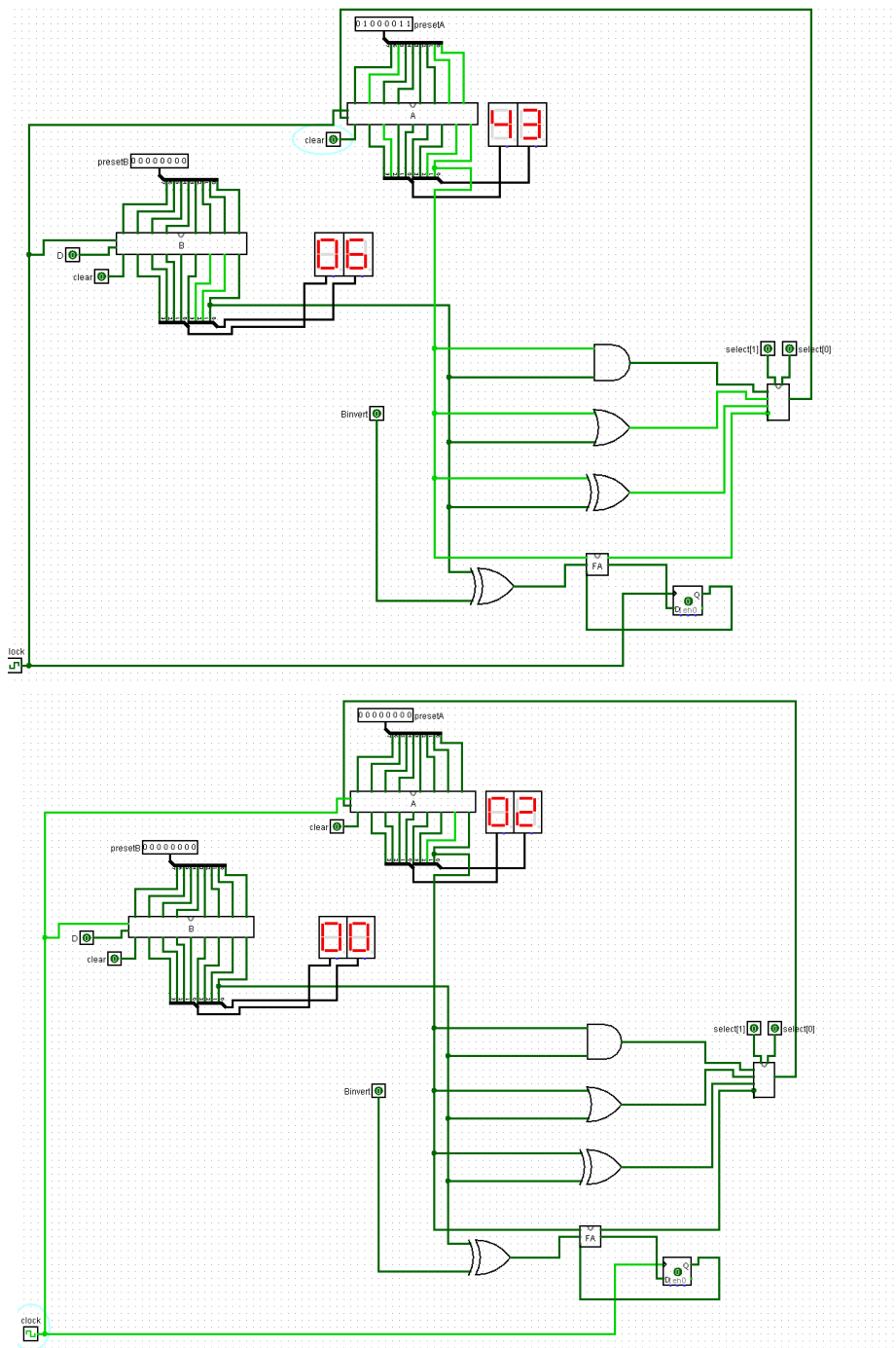
Para demonstrar a execução da operação NOT, testamos o funcionamento do MUX, visto que este depende de operações NOT internas. Definimos um par de operandos e os testamos com cada operação do MUX. Os resultados seguem abaixo.

AND:

RegistradorA = 01000011 (43)

RegistradorB = 00000110 (06)

Resultado esperado = 00000010 (2)



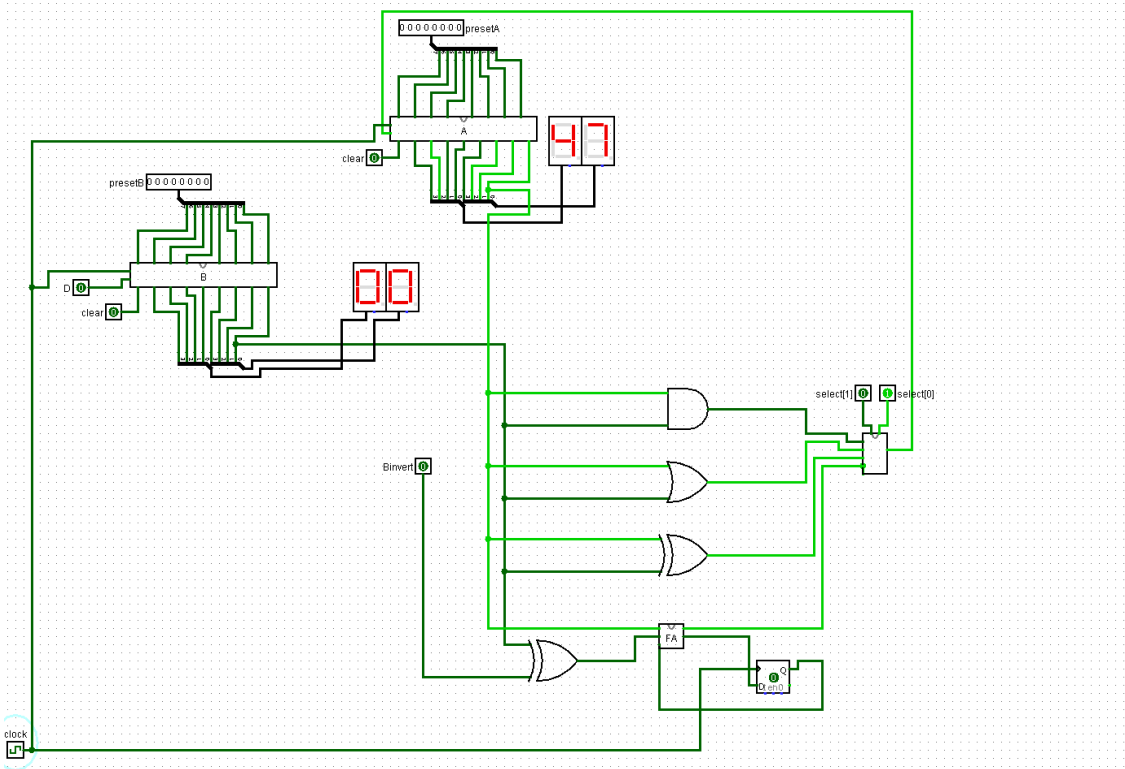
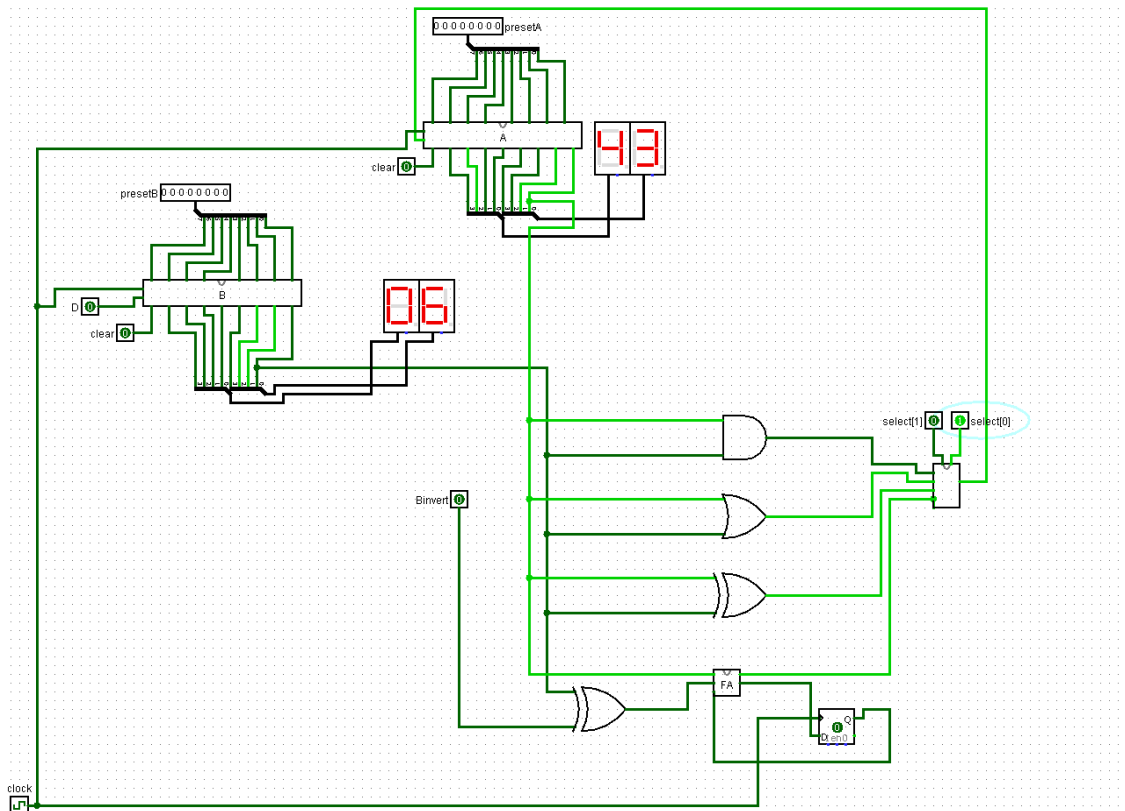
Atingiu-se o resultado esperado nesta operação.

OR:

RegistradorA = 01000011 (43)

RegistradorB = 00000110 (06)

Resultado esperado = 01000111 (47)



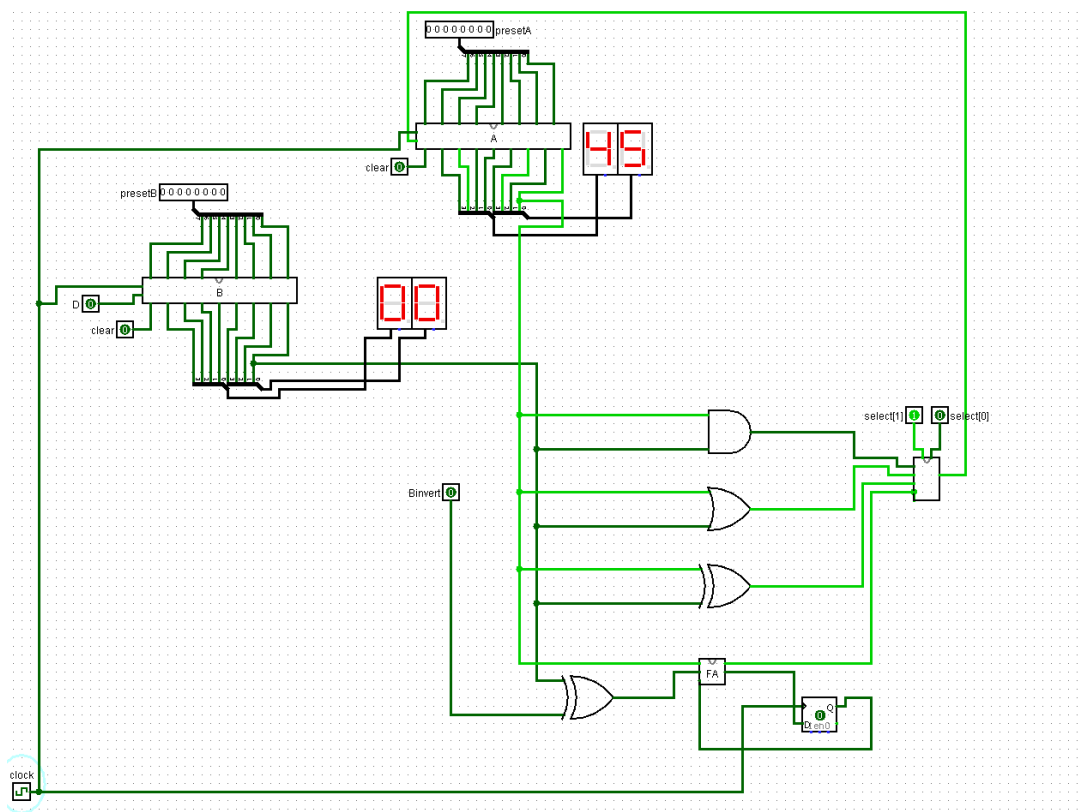
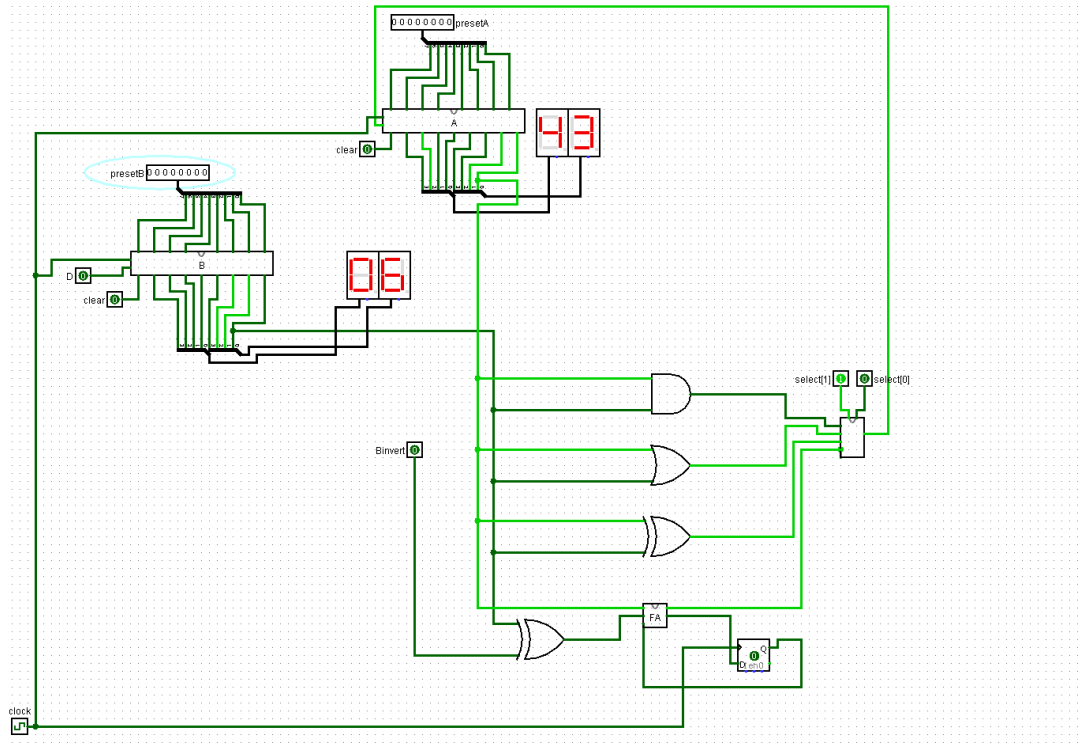
Atingiu-se o resultado esperado.

XOR:

RegistradorA = 01000011 (43)

RegistradorB = 00000110 (06)

Resultado esperado = 01000101 (45)



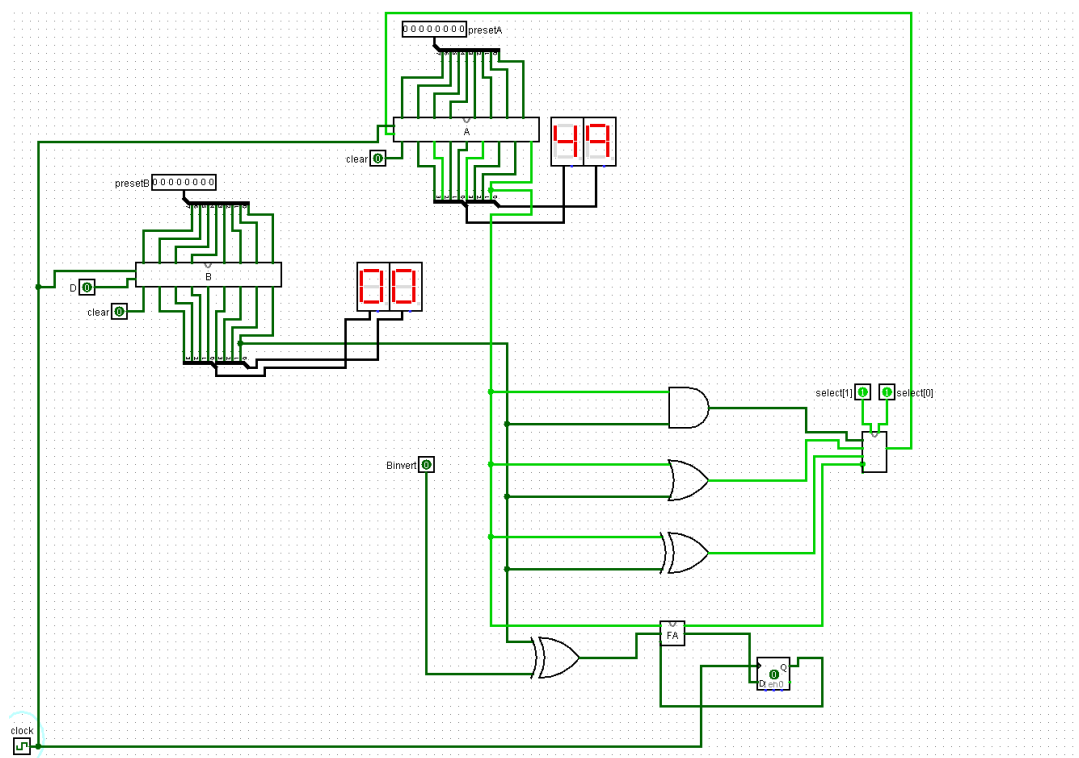
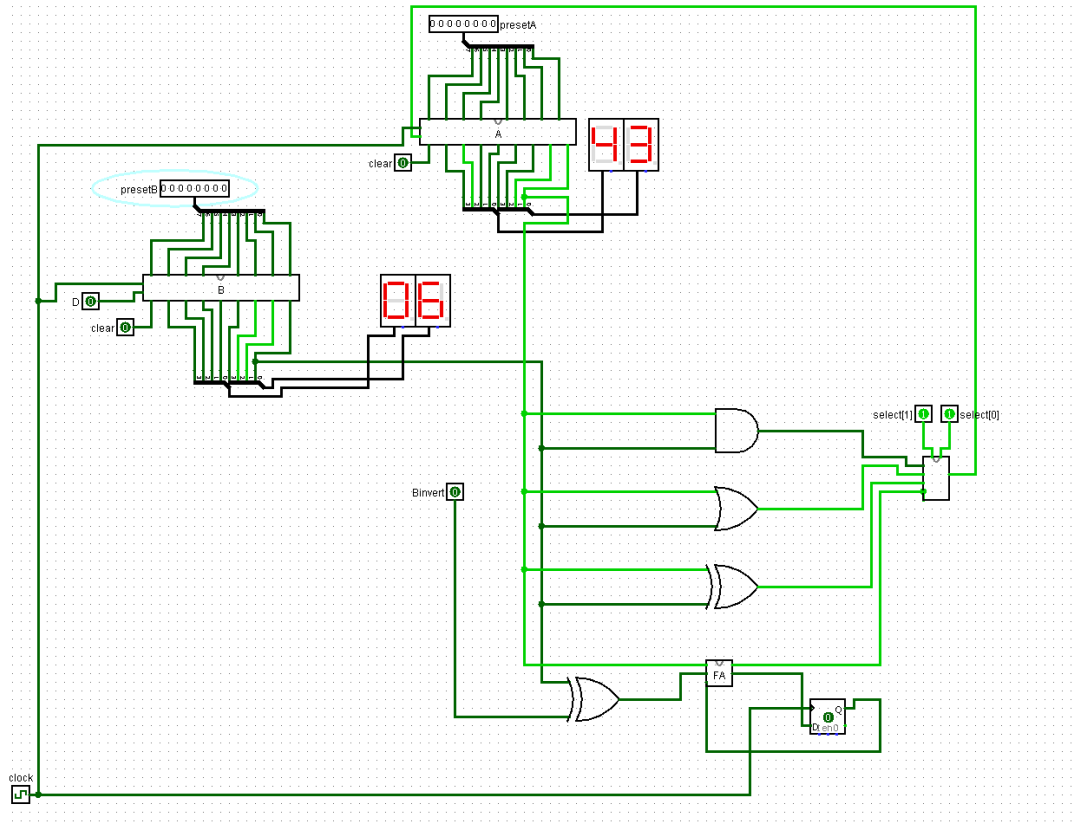
Atingiu-se o resultado esperado.

ADD:

RegistradorA = 01000011 (43)

RegistradorB = 00000110 (06)

Resultado esperado = 01001001 (49)



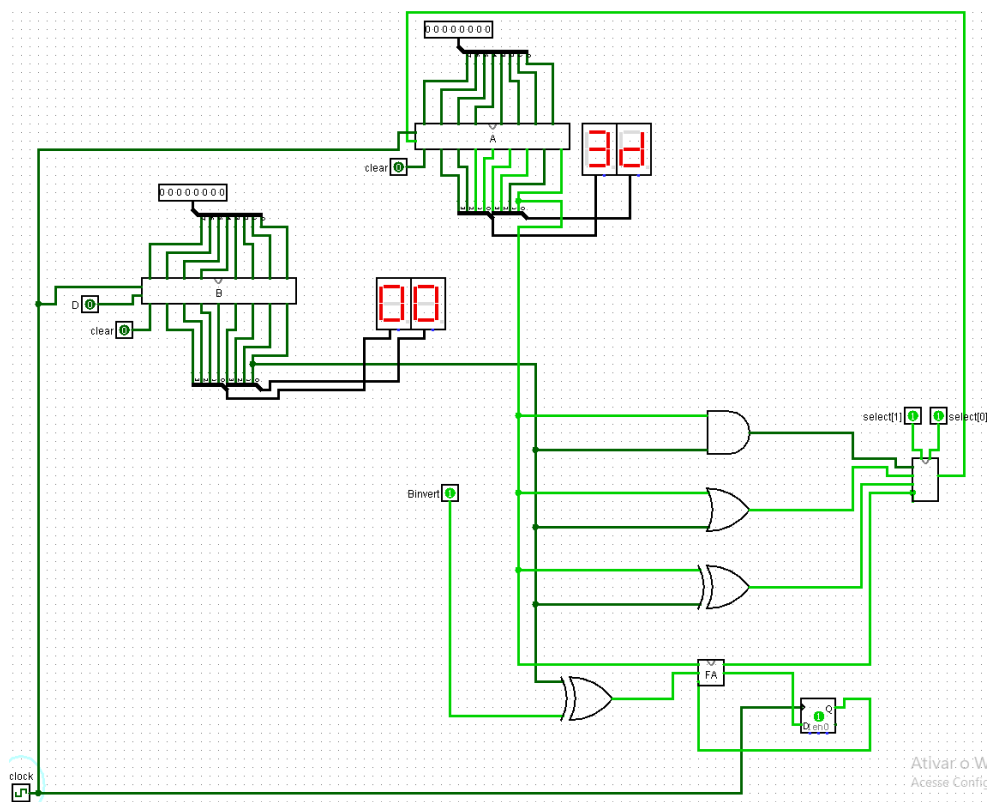
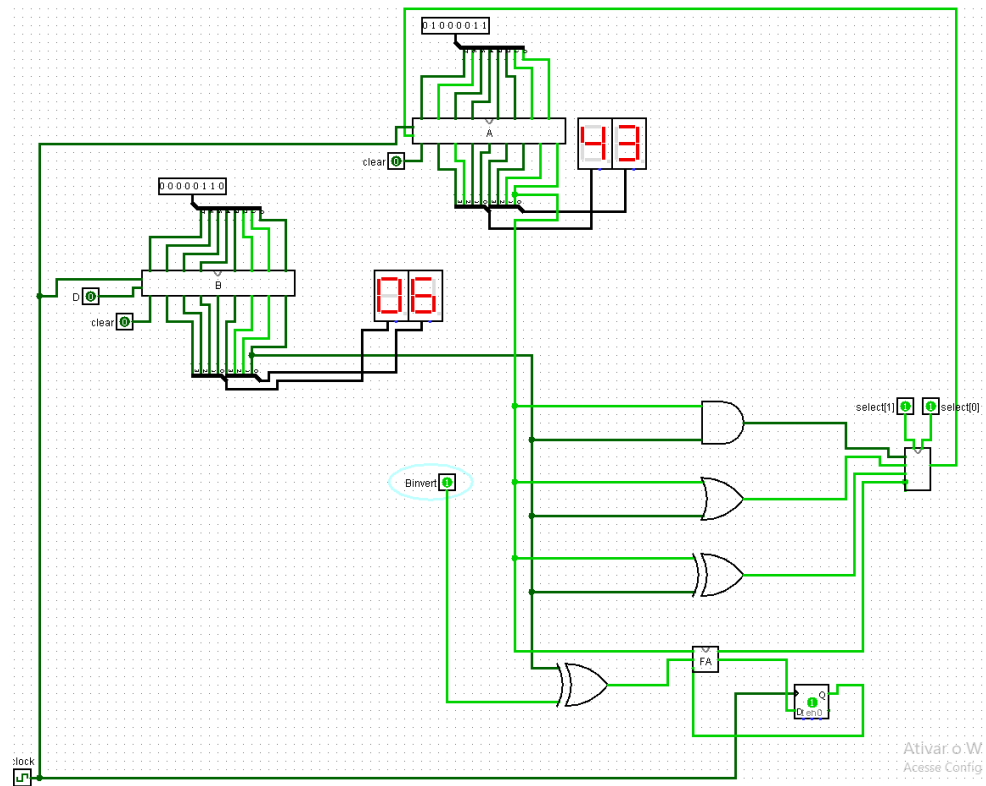
Atingiu-se o resultado esperado.

SUB:

RegistradorA = 01000011 (43)

RegistradorB = 00000110 (06)

Resultado esperado = 00111101 (3D)



Atingiu-se o resultado esperado.

XOR

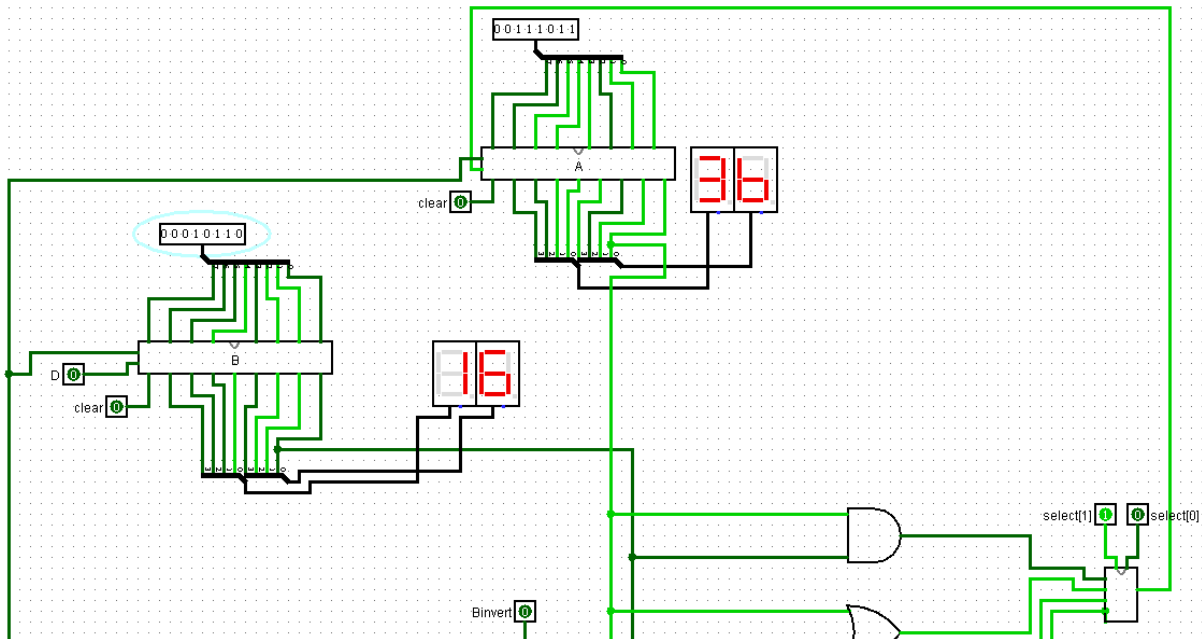
Ex.: 1

RegistradorA = 00111011 (3B)

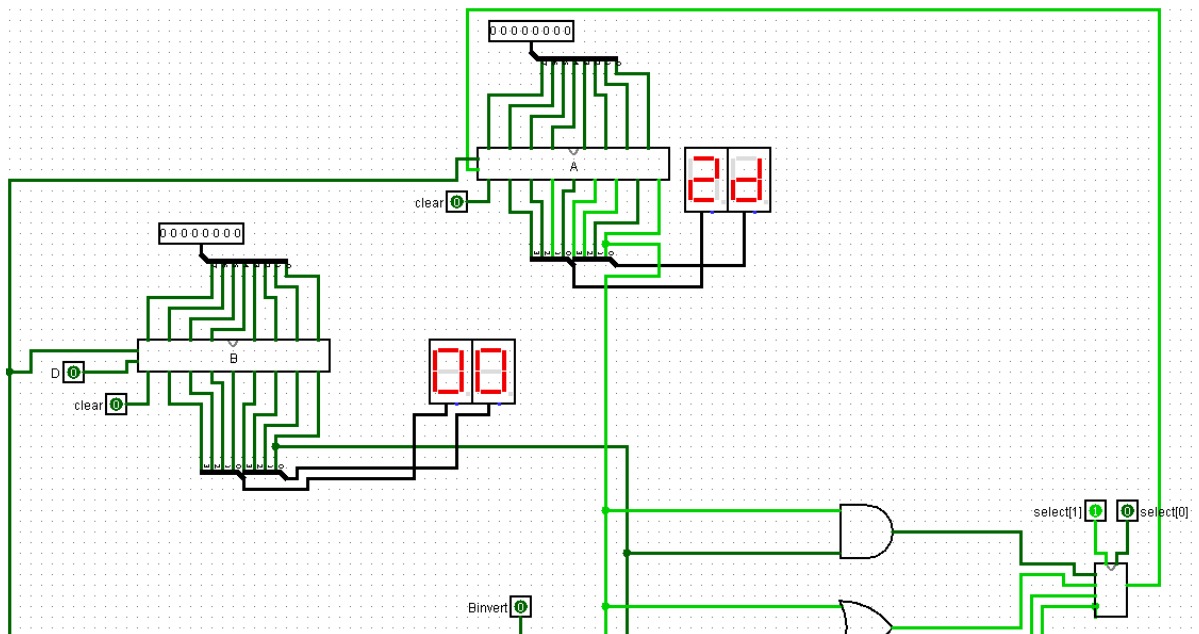
RegistradorB = 00010110 (16)

Resultado esperado = 00101101 (2D)

Seguindo o passo a passo da operação XOR, como foi escrito acima, a primeira ação é selecionar os valores desejados nos registradores, após isso, selecionar “1” no select[1] e “0” no select[0] do multiplexador.



Após zerar os números alterados e realizar os 8 pulsos do clock, o resultado é ilustrado no registradorA: 00101101 (2D)



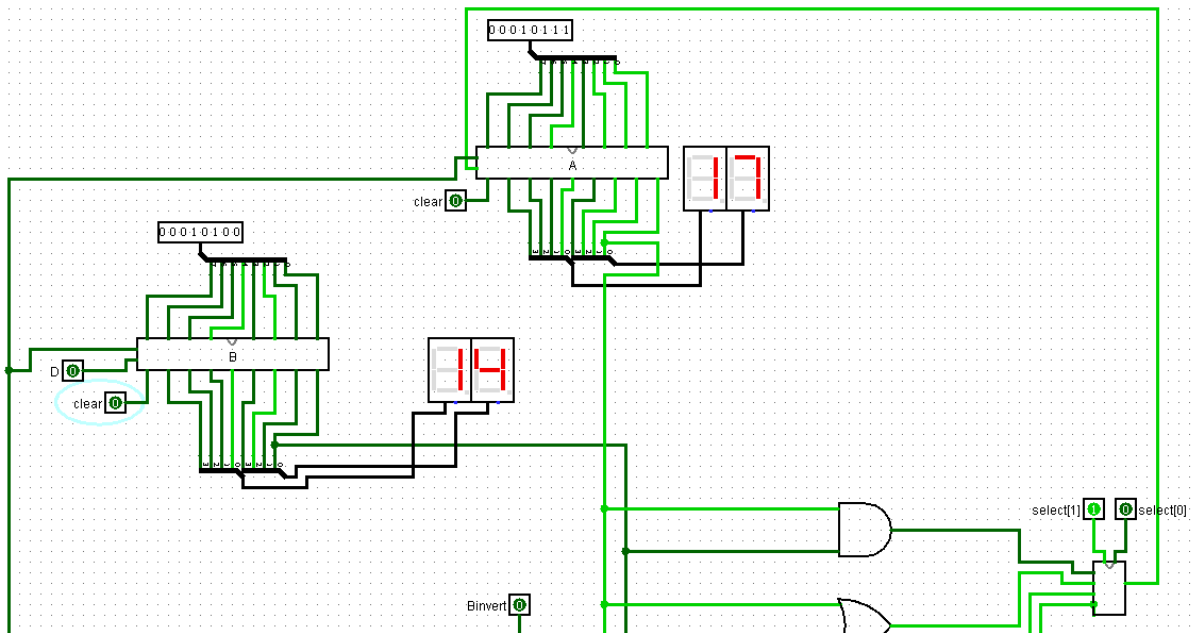
Ex.: 2

RegistradorA = 00010111 (17)

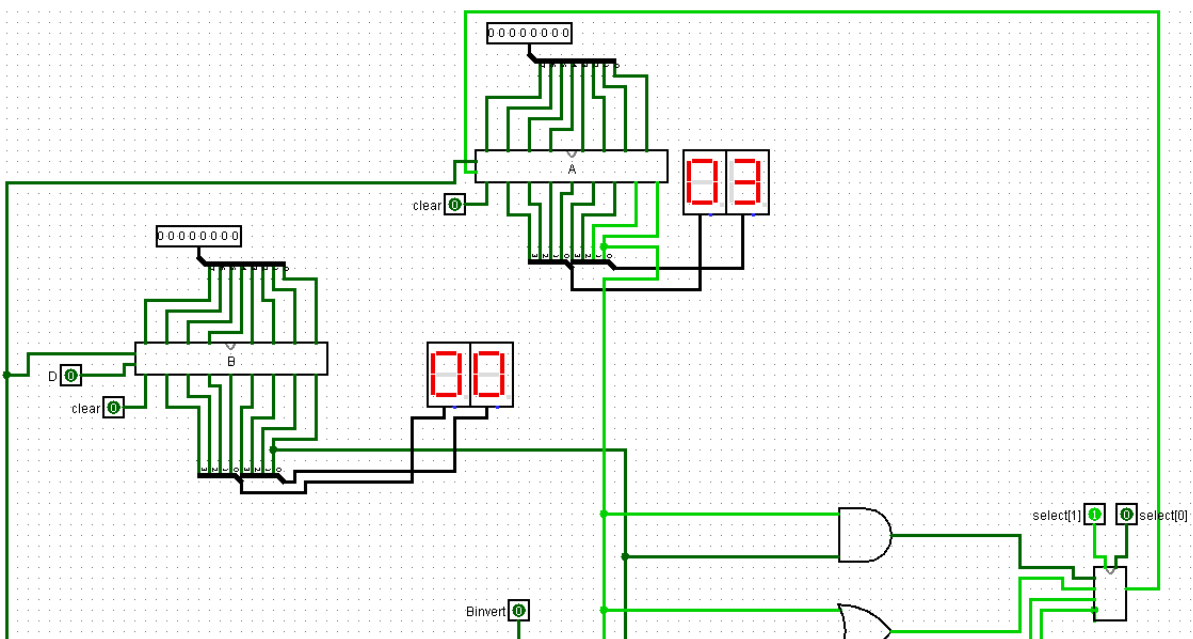
RegistradorB = 00010100 (14)

Resultado esperado = 00000011 (3)

Seguindo o passo a passo da operação XOR, como foi escrito acima, a primeira ação é selecionar os valores desejados nos registradores, após isso, selecionar “1” no select[1] e “0” no select[0] do multiplexador.



Após zerar os números alterados e realizar os 8 pulsos do clock, o resultado é ilustrado no registradorA: 00000011 (3)



ADD (adição)

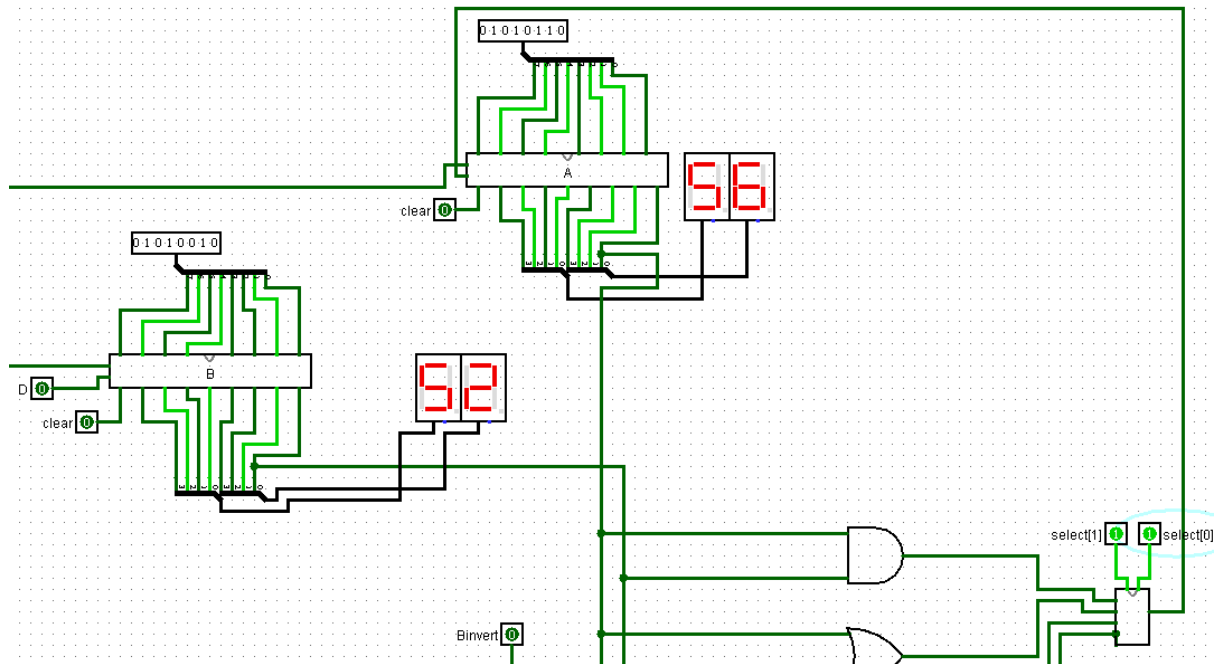
Ex.: 1

RegistradorA = 1010110 (56)

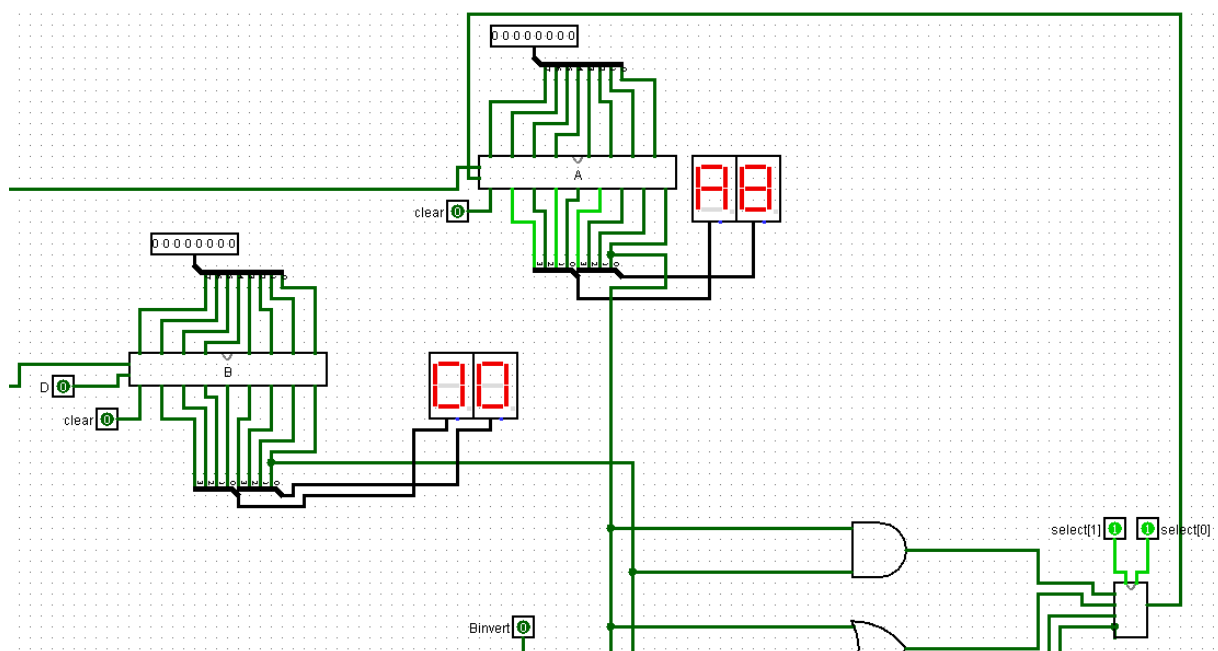
RegistradorB = 1010010 (52)

Resultado esperado = 10101000 (A8)

Seguindo o passo a passo da operação ADD, como foi escrito acima, a primeira ação é selecionar os valores desejados nos registradores, após isso, selecionar "1" tanto no select[1], quanto no select[0] do multiplexador.



Após zerar os números alterados e realizar os 8 pulsos do clock, o resultado é ilustrado no registradorA: 10101000 (A8)



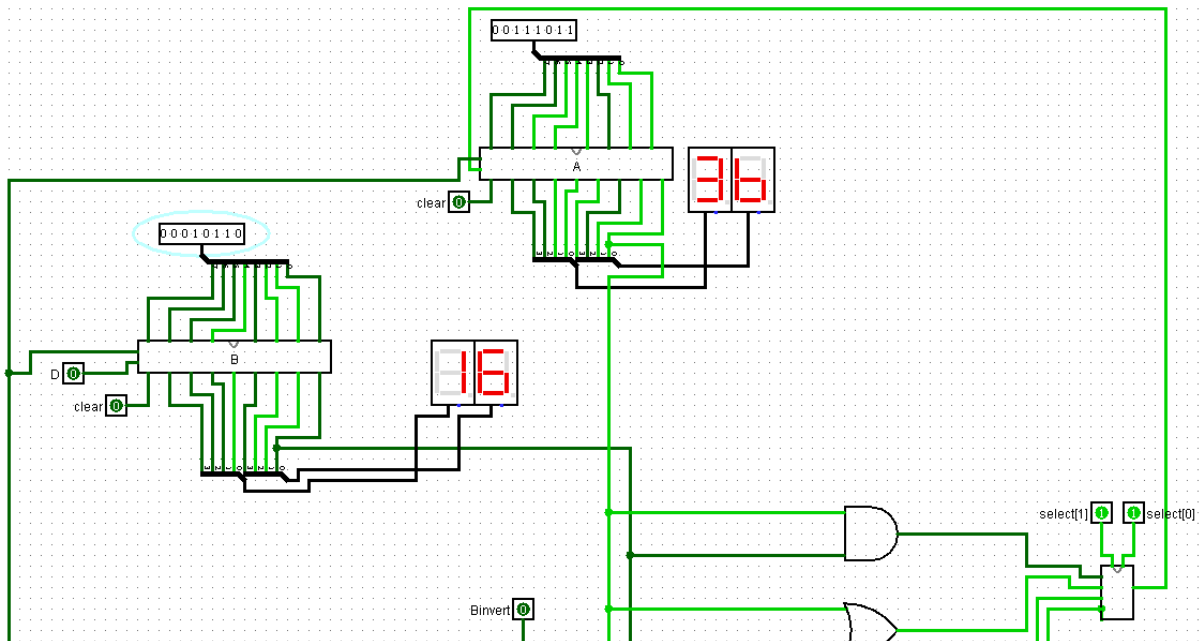
Ex.: 2

RegistradorA = 00111011 (3B)

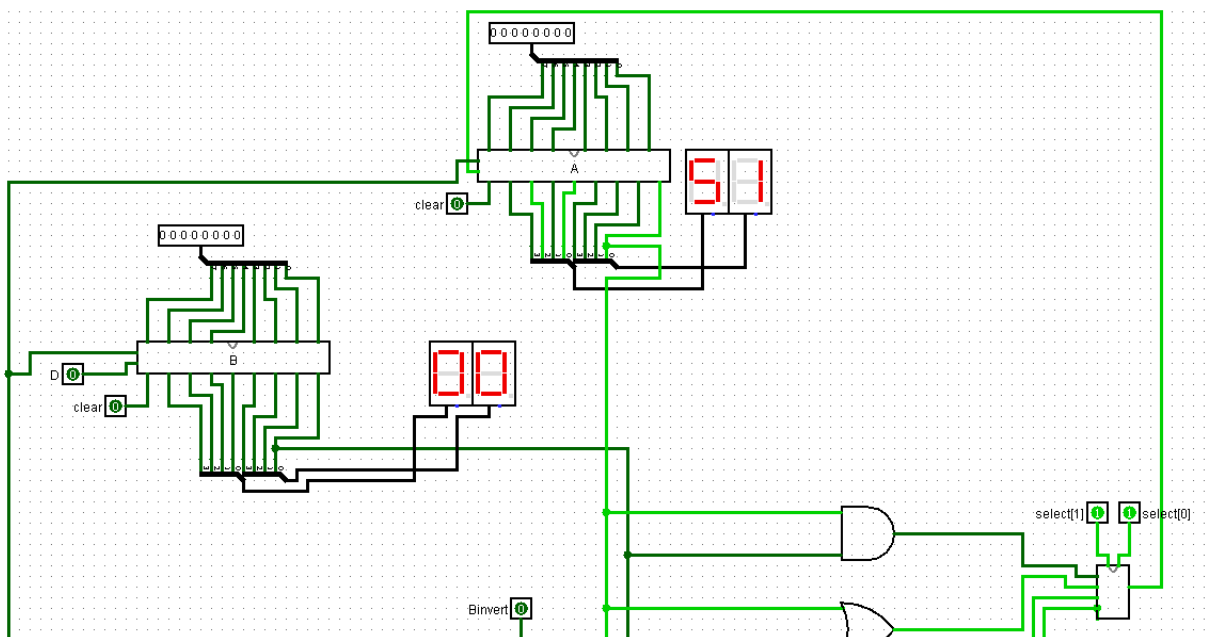
RegistradorB = 00010110 (16)

Resultado esperado: 01010001 (51)

Seguindo o passo a passo da operação ADD, como foi escrito acima, a primeira ação é selecionar os valores desejados nos registradores, após isso, selecionar “1” tanto no select[1], quanto no select[0] do multiplexador.



Após zerar os números alterados e realizar os 8 pulsos do clock, o resultado é ilustrado no registradorA: 01010001 (51)



SUB (subtração)

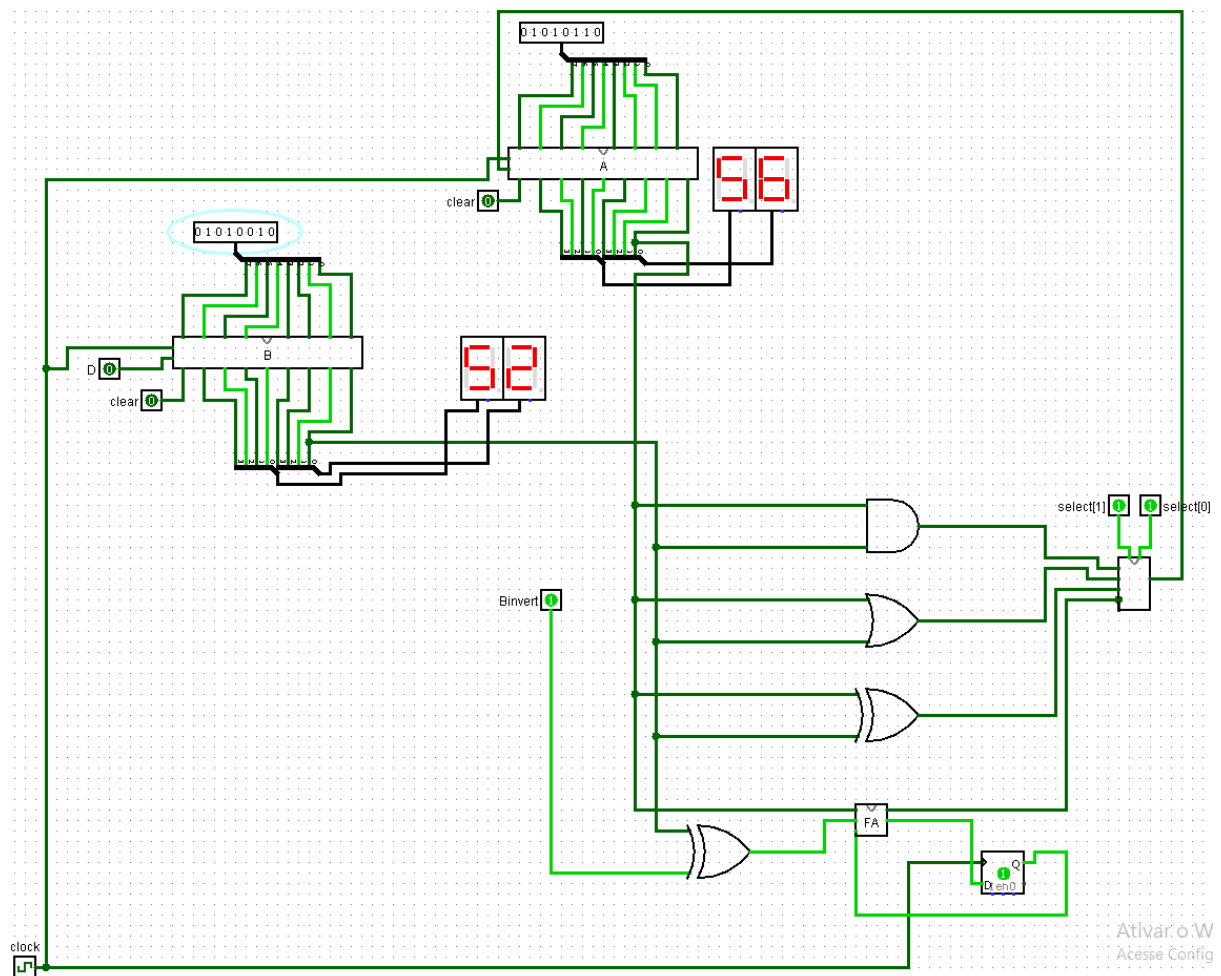
Ex.: 1

RegistradorA = 1010110 (56)

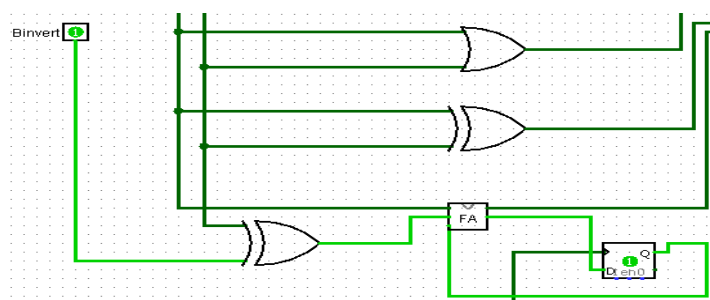
RegistradorB = 1010010 (52)

Resultado esperado = 00001000 (4)

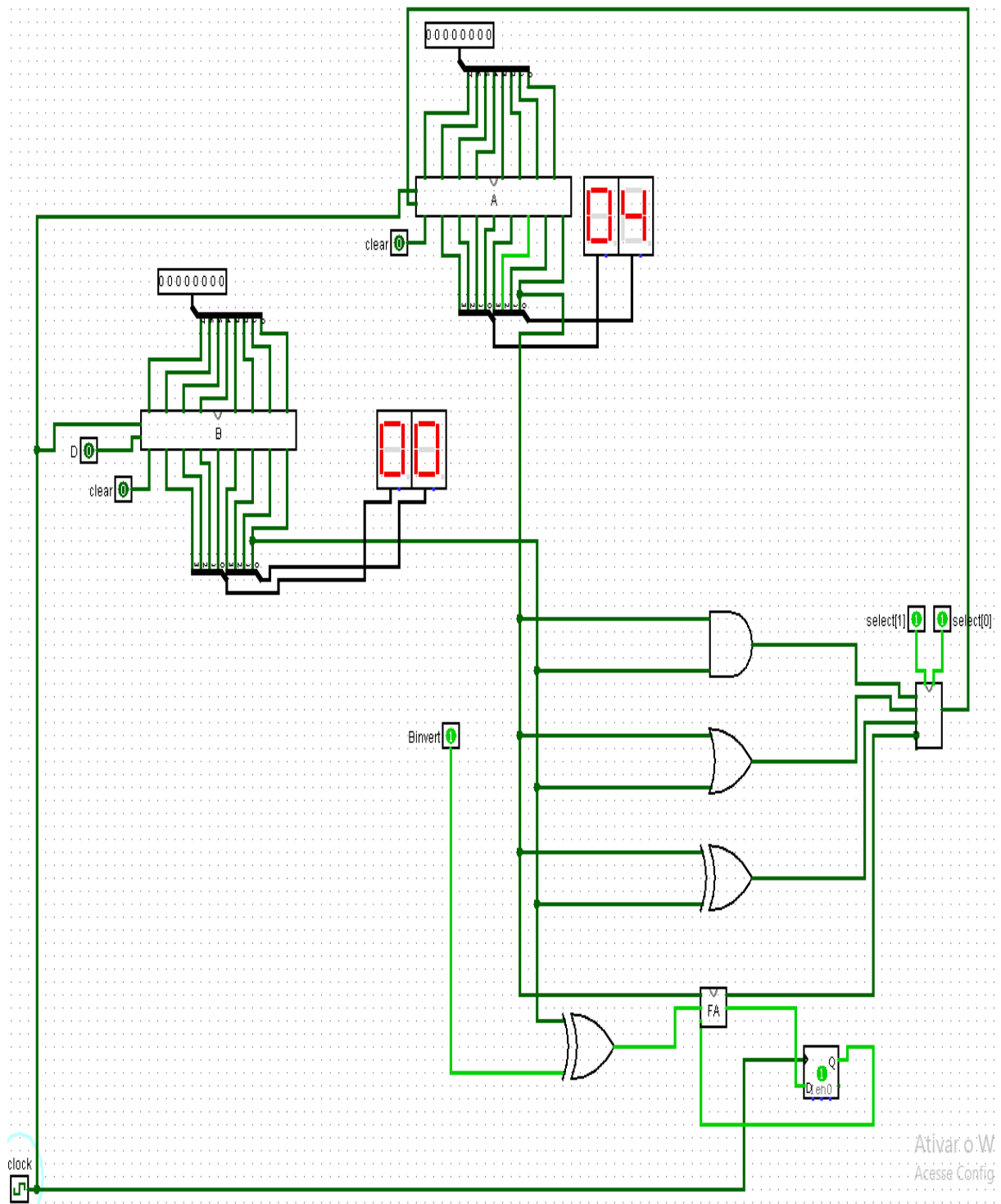
Seguindo o passo a passo da operação SUB, como foi escrito acima, a primeira ação é selecionar os valores desejados nos registradores, após isso, selecionar "1" tanto no select[1], quanto no select[0] do multiplexador. Por fim, selecionar "1" no Binvert e no Carry para o circuito entender que trata-se de uma subtração.



Note que o valor selecionado no multiplexador é o mesmo selecionado para realizar uma adição, o que faz o circuito entender que trata-se de uma subtração é a mudança no Binvert e no Carry.



Após zerar os números alterados e realizar os 8 pulsos do clock, o resultado é ilustrado no registradorA: 00001000 (4)



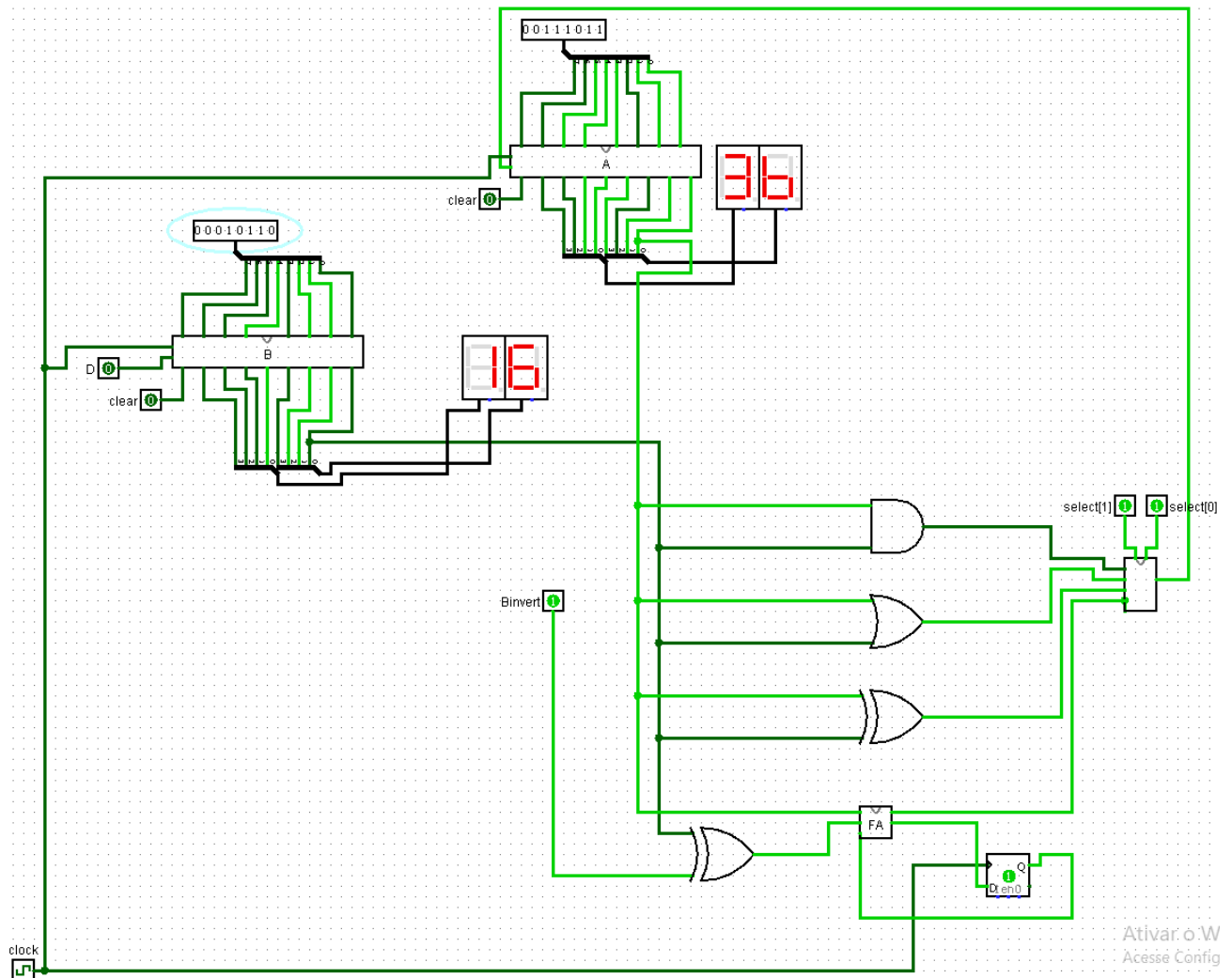
Ex.: 2

RegistradorA = 00111011 (3B)

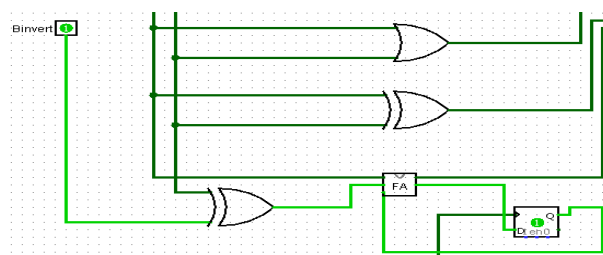
RegistradorB = 00010110 (16)

Resultado esperado = 00100101 (25)

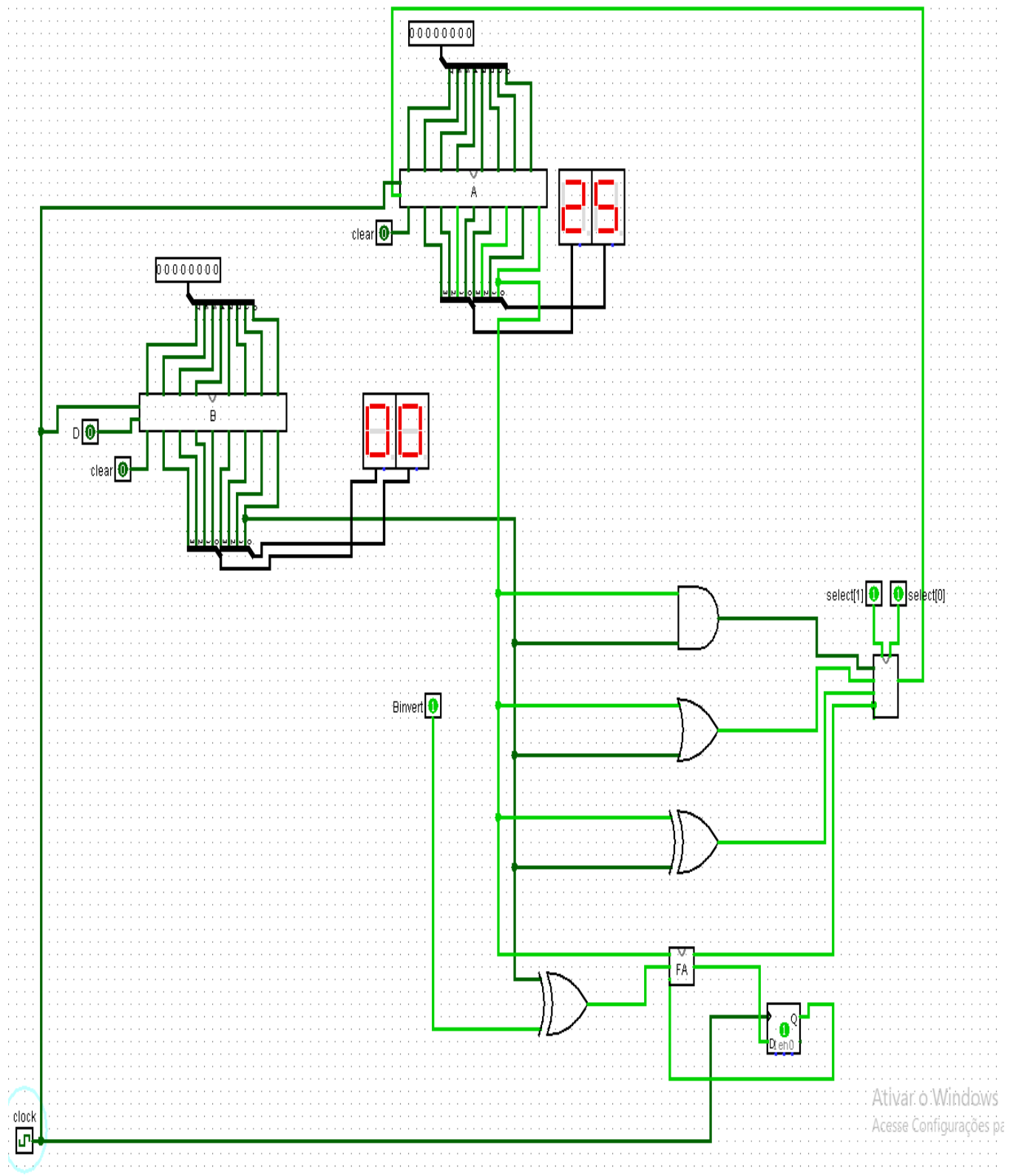
Seguindo o passo a passo da operação SUB, como foi escrito acima, a primeira ação é selecionar os valores desejados nos registradores, após isso, selecionar “1” tanto no select[1], quanto no select[0] do multiplexador. Por fim, selecionar “1” no Binvert e no Carry para o circuito entender que trata-se de uma subtração.



Note que o valor selecionado no multiplexador é o mesmo selecionado para realizar uma adição, o que faz o circuito entender que trata-se de uma subtração é a mudança no Binvert e no Carry.



Após zerar os números alterados e realizar os 8 pulsos do clock, o resultado é ilustrado no registradorA: 00100101 (25)



Observações:

Com a realização e demonstração dos mais diversos valores em cada operação, conclui-se que a ULA funciona perfeitamente se for configurada do jeito correto. Em questão dos valores utilizados, alguns valores eram semelhantes em diferentes operações, isso é para mostrar que o resultado é diferente em cada operação. Não obstante, os operandos não foram repetidos em todas as operações para também mostrar que nos mais diferentes casos a operação também daria certo.

- Foram usados os mesmos operandos entre os exemplos das operações AND e OR;
- Foram usados os mesmos operandos entre os exemplos das operações ADD e SUB;
- Os operandos do exemplo 2 do XOR foram tirados do exemplo 1 do AND e OR
- Os operandos do exemplo 1 do XOR foram tirados do exemplo 2 do ADD e SUB;
- Todos os operandos utilizados dentro do exemplo de uso da operação NOT são iguais, mas não é repetido em nenhuma outra operação.