

”

**E-fólio A** | Folha de resolução para E-fólio

**UNIDADE CURRICULAR:** Arquitetura de Computadores

**CÓDIGO:** 21010

**DOCENTE:** Gracinda Carvalho; José Coelho

**A preencher pelo estudante**

**NOME:** JOSÉ AUGUSTO OLIVEIRA AZEVEDO

**N.º DE ESTUDANTE:** 2200655

**CURSO:** LICENCIATURA EM ENGENHARIA INFORMÁTICA

## TRABALHO / RESOLUÇÃO:

### Alínea A

**Criar um componente para descodificar o código de uma carta e mostrar a carta em dois displays de 7 segmentos.**

Informação do Circuito combinatório:

- Entradas:
  - o Naipes: 2 bits – naipe codificado
  - o Número: 2 bits – número codificado
  - o EN: 1 bit – sinal de enable (não mostrar nada se inativo)
- Sidas:
  - o Número: 7 bits – número descodificado
  - o Naipes: 7 bits – naipe descodificado

Analisando o problema relativamente aos Naipes, temos 2 bits de entrada que podem ser 00, 01, 10 e 11 a que corresponde, respectivamente, a saída P, O, C, E apresentada no display de 7 segmentos. Em relação ao número, temos 2 bits de entrada que podem ser 00, 01, 10 e 11 a que corresponde, respectivamente, a saída A, 1, 2, 3 apresentada no display de 7 segmentos.

Isto pode ser traduzido na tabela da verdade que representa as situações em que o enable/control está ativo e se encontra no anexo **Figura 1 – Alínea A – Tabela da verdade com o enable/control ativo.**

Elaborei os mapas de Karnaugh para os naipes e para os números e cada uma das saídas (ver anexo, **Figura 2 – Alínea A – Mapas de Karnaugh dos Naipes** e **Figura 3 – Alínea A – Mapas de Karnaugh dos Números**) e obtive as expressões que constam da **Figura 4 – Alínea A – Expressões resultantes dos Mapas de Karnaugh.**

Criei no Digital Works o circuito digital com as expressões obtidas anteriormente (ver anexo, **Figuras 5, 6, 7 e 8 – Alínea A – Circuito lógico – Com teste de entradas**). Após efetuar todos os testes criei o componente, ver anexo **Figura 9 - Alínea A – Criação do Componente.** Criei circuito com o componente e efetuei os testes solicitados, ver anexo **Figura 10 e Figura 11 – Alínea A – Utilização do Componente – Com teste de entradas.**

Ficheiros:

2200655\_21010\_A\_01\_CircuitoBase.dwm - Figuras 5, 6, 7 e 8

2200655\_21010\_A\_02\_Componente.dwm – Figura 9

2200655\_21010\_A\_03\_UtilizacaoComponente.dwm - Figura 11

## Alínea B

**Criar um componente que permita seleccionar uma de quatro linhas e uma de quatro colunas, e mostre ao utilizador a carta nessa posição.**

Em primeiro lugar elaborei uma tabela que traduzia a selecção de uma linha e uma coluna e apresentava na saída qual o endereço de memória a consultar, ver anexo **Figura 12 – Alínea B – Mapeamento Linha/Coluna Saída**.

Analizando a tabela rapidamente conclui que era relativamente fácil identificar qual a dependência que cada bit de saída tinha em relação às entradas:  $S_0 = C_1 + C_3$ ;  $S_1 = C_2 + C_3$ ;  $S_2 = L_1 + L_3$ ;  $S_3 = L_2 + L_3$ ; ver anexo **Figura 13 – Alínea B – Expressões lógicas das saídas**.

Também era necessário identificar quando é que a entrada de dados estava válida, ou seja, quando é que estava seleccionada apenas uma linha e uma coluna. Assim elaborei uma tabela da verdade do enable ou dados válidos, ver anexo **Figura 14 – Alínea B – Calculo do enable ou de selecção válida das linhas/colunas**. Desta tabela obtivo uma soma de produtos para validar as colunas e outra soma de produtos para validar as linhas.

Olhando para as expressões obtidas na primeira tabela temos que a saída  $S_0$  é idêntica à saída  $S_2$  e a saída  $S_1$  é idêntica à saída  $S_3$ , usam-se sempre os mesmos índices das entradas, nos primeiros casos as colunas e nos segundos as linhas. As expressões de validação das entradas (uma única seleccionada) também são iguais, usando as colunas no primeiro caso e as linhas no segundo.

Significa isto que é suficiente criar um componente com duas entradas e duas saídas para traduzir a selecção de 1 de 4 bits de entrada para os 2 bits de saída e depois duplicar a utilização desse componente, uma vez para a selecção de linhas e outra vez para a selecção de colunas.

Foi criado o circuito lógico em que se selecciona uma entrada de 4 (escolha da linha ou da coluna) e se coloca na saída os bits respectivos; ver anexo: **Figura 15 – Alínea B – Circuito lógico**. Depois foi criado o componente referente ao circuito anterior (**Figura 16 – Alínea B – Componente com duas entradas e duas saídas**). O passo seguinte foi criar o circuito lógico para responder à alínea B, utilizando duas vezes o componente anterior, ver **Figura 17 – Alínea B – Circuito lógico utilizando 2 componentes**, depois criar o componente final apartir do circuito **Figura 18 – Alínea B – Componente final**. Para finalizar foram efetuados testes utilizando o componente que permite ver anexo **Figura 19 – Alínea B – Utilização do Componente – Com testes de entradas**.

## Alínea C

**Fazer um circuito sequencial em que escolha em sequência duas cartas, sem revelar a primeira carta até que a segunda seja escolhida. Após a escolha de ambas as cartas, estas devem ser mostradas em simultâneo, até que nova carta seja escolhida.**

Ao analisar o problema, decidi que para perceber em que passo estava da sequência ia utilizar um contador de 2 bits que elaborei com flip-flops tipo D e criei um componente, ver anexo **Figura 20** – Alínea C – Componente contador. Também reparei que os passos dependem de quando é que as entradas estão válidas ou não, pelo que alterei o componente criado na Alínea B por forma a que tenho na saída um indicador da validade das entradas. (**Figura 21** – Alínea C – Componente Aline B com output de entradas válidas).

Em seguida, avancei para a questão de armazenar os valores das cartas, já que só são apresentados no final, pelo que criei um componente de memória de 7 bits usando flip-flops tipo D (**Figura 22** – Alínea C – Componente Memória 7 bits). No circuito iria utilizar 4 componentes de memória (um por display) pelo que teria que conseguir direccionar a saída do Componente da Alinea B (com a alteração) para a memória respetiva, usando como base em que passo da sequência é que o circuito se encontrava (contador). Assim, elaborei um Multiplexador de Selecção que permite escolher em que pares de memórias é que a informação é guardada utilizando dois selectores (**Figura 23** – Alínea C – Componente Multiplexador de Selecção). Para finalizar foi necessário criar um buffer que só permitisse a passagem dois bits da memória para os displays quando o passo 5 fosse atingido (**Figura 24** – Alínea C – Componente Buffer).

De referir que foi essencial perceber algumas questões como em que situações deveria o reset do contador ser ativado para reiniciar a sequência; a selecção do multiplexador de acordo com o contador e quando é que o buffer fica ativo; ver anexo, **Figura 25** – Alínea C – Estudo e validações. Com estes componentes efetuei a montagem do circuito (ver anexo **Figura 26** – Alínea C – Circuito).

Para finalizar efetuei diversos testes, ver anexo, da Figura 27 até Figura 36 (Alínea C – Circuito com testes).

## Alínea D

O primeiro passo que efetuei para a resolução desta alínea foi alterar o componente que valida a seleção inicial a apresentar no output o endereço de memória que corresponde à selecção efetuada também foi colocado uma entrada de enable, ver anexo **Figura 37** – Alínea D – Componente com output de entradas válidas e endereço de memória. Depois criei um componente de memória de 16 bits (**Figura 38** – Alínea D – Componente memória 16 bits). Neste componente cada bit corresponde a uma posição de memória das cartas e quando as cartas são para remover a posição respetiva é colocada a 1, basicamente funciona como um array onde cada posição corresponde a um endereço da memória e quando temos elementos removidos o valor é colocado a 1, significando que já não é válido. Assim foi criado um componente que converte o endereço de memória para a posição no array, ver anexo **Figura 39** – Alínea D – Componente conversor Endereço Memória para posição no array.

Da mesma forma que na alínea anterior, era importante preservar o endereço de memória das cartas seleccionadas pelo que foi criado um componente de memória de 4 bits e também foi criado o selector inicial, ver **Figura 40** – Alínea D – Componente Memória bits e **Figura 41** – Alínea D – Componente Selector inicial. Para juntar o valor de ambas as selecções (posições no array) foi criado um componente que efetua o OR dos array. (**Figura 43** – Alínea D – Componente Passagem OR Array)

Para verificar se o valor das cartas ou os naipes eram idênticos foi criado um Componente comparador de 7 bits (**Figura 42** – Alínea D – Componente Comparador de 7 bits). Existindo cartas a remover no circuito é utilizado um componente que escreve na memória de 16 bits (ver anexo, **Figura 44** – Alínea D – Componente Escrita Array).

Foram criados dois contadores (o de jogadas, **Figura 45** – Alínea D – Componente Contador 8 bits) e o contador que efetua a contagem dos pares, **Figura 46** – Alínea D – Componente Contador até 8 (pares).

O circuito ao atingir o número de 8 pares não permite que sejam efetuadas mais jogadas. Foi criado um componente que valida as entradas em relação ao Array (ver anexo, **Figura 47** – Alínea D – Componente valida entrada em relação ao Array). No anexo são apresentadas várias vistas parciais do circuito final desta alínea.

Foram efetuados vários testes e apresento um teste de jogo completo no anexo:

### Teste de jogo completo

Conteúdo ficheiro de memória efolioA0.map

Memória		
Endereço	Valor	Carta
0000	1001	2O
0001	0011	AE
0002	1000	2P
0003	0000	AP
0004	0001	AO
0005	1010	2C
0006	1110	3C
0007	0111	1E
0008	0101	1O
0009	1111	3E
000A	0100	1P
000B	1011	2E
000C	1101	3O
000D	1100	3P
000E	0010	AC
000F	0110	1C

Bits	Valor	Naípe
00	A	P
01	1	O
10	2	C
11	3	E

Matriz	4	3	2	1
1	AP	2P	AE	2O
2	1E	3C	2C	AO
3	2E	1P	3E	1O
4	1C	AC	3P	3O

Matriz	4	3	2	1
1	AP	2P	AE	2O
2	1E	3C	2C	AO
3	2E	1P	3E	1O
4	1C	AC	3P	3O

Matriz (Ic)	Carta
14	AP
42	3P
34	2E
12	AE
24	1E
32	3E
13	2P
33	1P
44	1C
43	AC
23	3C
22	2C
11	2O
31	1O
21	AO
41	3O

As jogadas foram efetuadas pelas ordem da tabela à direita.

Ver anexo da Figura 51 à Figura 89.

## ANEXOS:

Entradas					Saída						
Naipes	Valor	a	b	control	1	2	3	4	5	6	7
Paus	0	0	0	1	1	1	0	0	1	1	1
Ouros	1	1	0	1	1	1	1	1	1	1	0
Copas	2	0	1	1	1	0	0	1	1	1	0
Espadas	3	1	1	1	1	0	0	1	1	1	1

Entradas					Saída						
Número	Valor	d	e	control	1	2	3	4	5	6	7
Às	0	0	0	1	1	1	1	0	1	1	1
Dois	1	1	0	1	0	1	1	0	0	0	0
Três	2	0	1	1	1	1	0	1	1	0	1
Quatro	3	1	1	1	1	1	1	1	0	0	1

**Figura 1 – Alínea A – Tabela da verdade com o enable/control ativo**

ab	Naipes - Saída 1					ab	Naipes - Saída 5				
c	00	01	11	10		c	00	01	11	10	
1	1	1	1	1	c	1	1	1	1	1	c
0	0	0	0	0		0	0	0	0	0	

ab	Naipes - Saída 2					ab	Naipes - Saída 6				
c	00	01	11	10		c	00	01	11	10	
1	1	0	0	1	/bc	1	1	1	1	1	c
0	0	0	0	0		0	0	0	0	0	

ab	Naipes - Saída 3					ab	Naipes - Saída 7				
c	00	01	11	10		c	00	01	11	10	
1	0	0	0	1	a/bc	1	1	0	1	0	/a/bc+abc
0	0	0	0	0		0	0	0	0	0	

ab	Naipes - Saída 4										
c	00	01	11	10							
1	0	1	1	1	bc+ac						
0	0	0	0	0							

**Figura 2 – Alínea A – Mapas de Karnaugh dos Naipes**

de	Número - Saída 1					de	Número - Saída 5				
c	00	01	11	10		c	00	01	11	10	
1	1	1	1	0	/dc+ec	1	1	1	0	0	/dc
0	0	0	0	0		0	0	0	0	0	p

de	Número - Saída 2					de	Número - Saída 6				
c	00	01	11	10		c	00	01	11	10	
1	1	1	1	1	c	1	1	0	0	0	/d/ec
0	0	0	0	0		0	0	0	0	0	

de	Número - Saída 3					de	Número - Saída 7				
c	00	01	11	10		c	00	01	11	10	
1	1	0	1	1	dc+ec	1	1	1	1	0	/dc+ec
0	0	0	0	0		0	0	0	0	0	

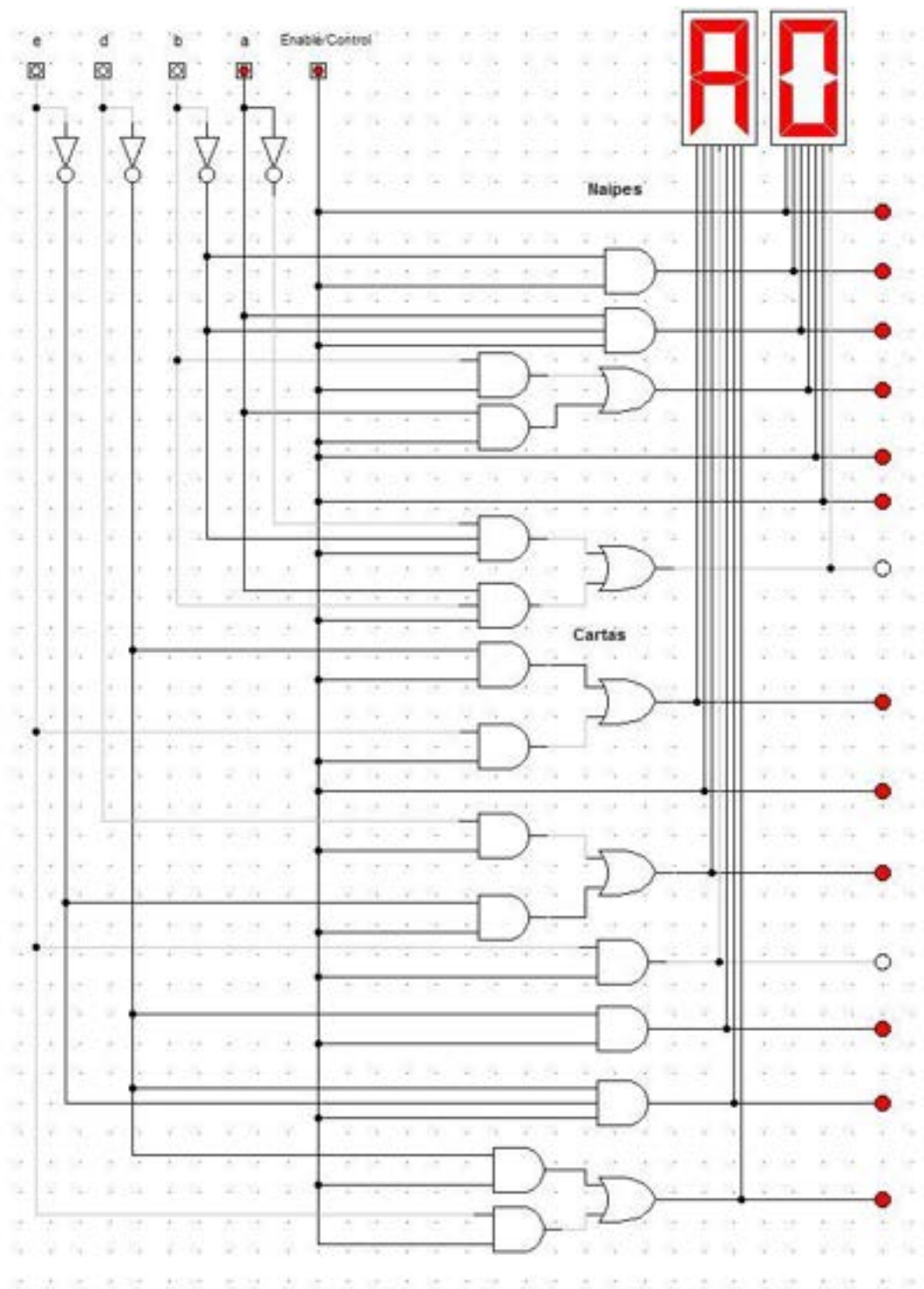
  

de	Número - Saída 4										
c	00	01	11	10							
1	0	1	1	0	ec						
0	0	0	0	0							

**Figura 3 – Alínea A – Mapas de Karnaugh dos Números**

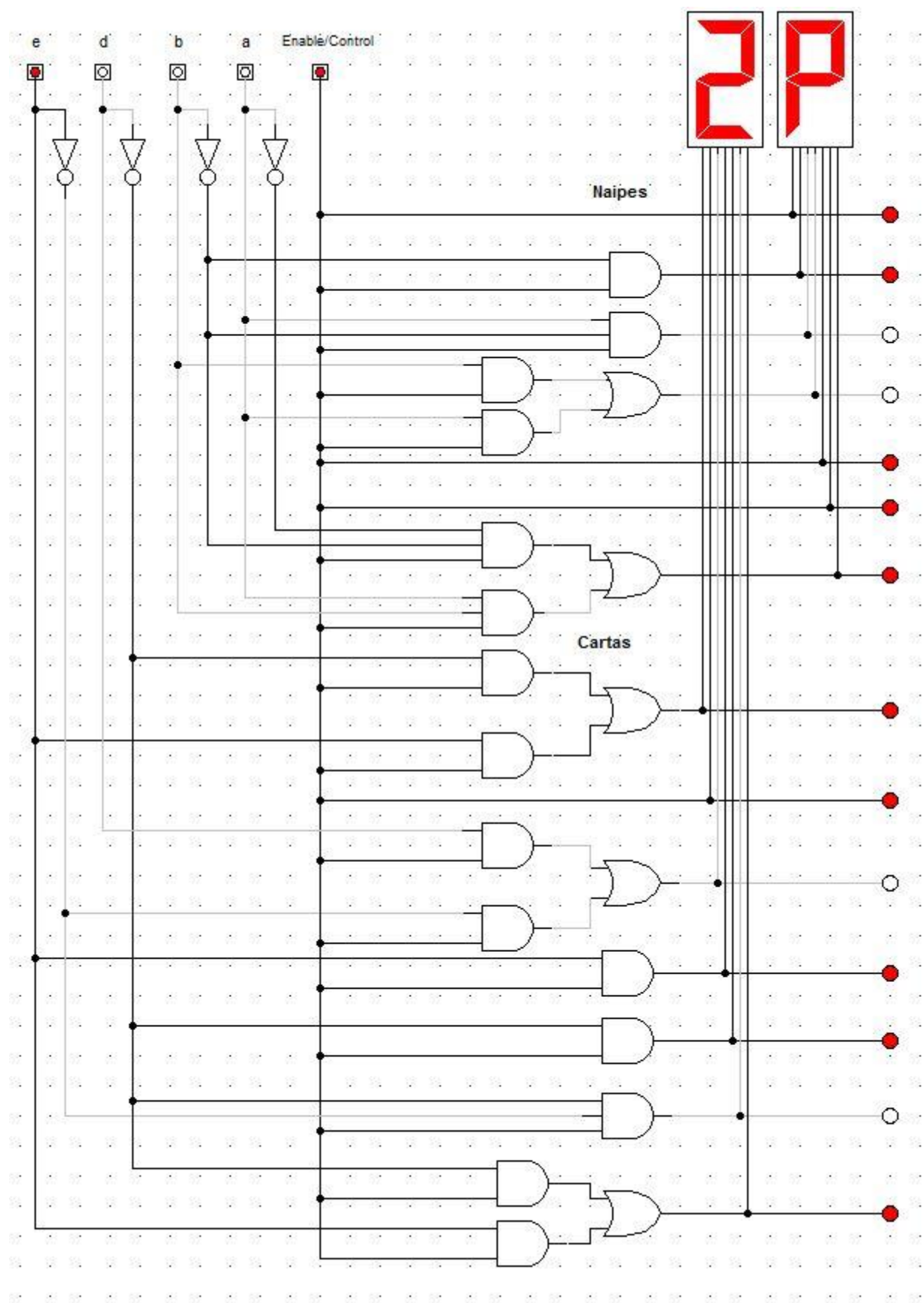
Naipes		Números	
Saída 1	c	Saída 1	/dc+ec
Saída 2	/bc	Saída 2	c
Saída 3	a/bc	Saída 3	dc+ec
Saída 4	bc+ac	Saída 4	ec
Saída 5	c	Saída 5	dc
Saída 6	c	Saída 6	/d/ec
Saída 7	/a/bc+abc	Saída 7	/dc+ec

**Figura 4 – Alínea A – Expressões resultantes dos Mapas de Karnaugh**

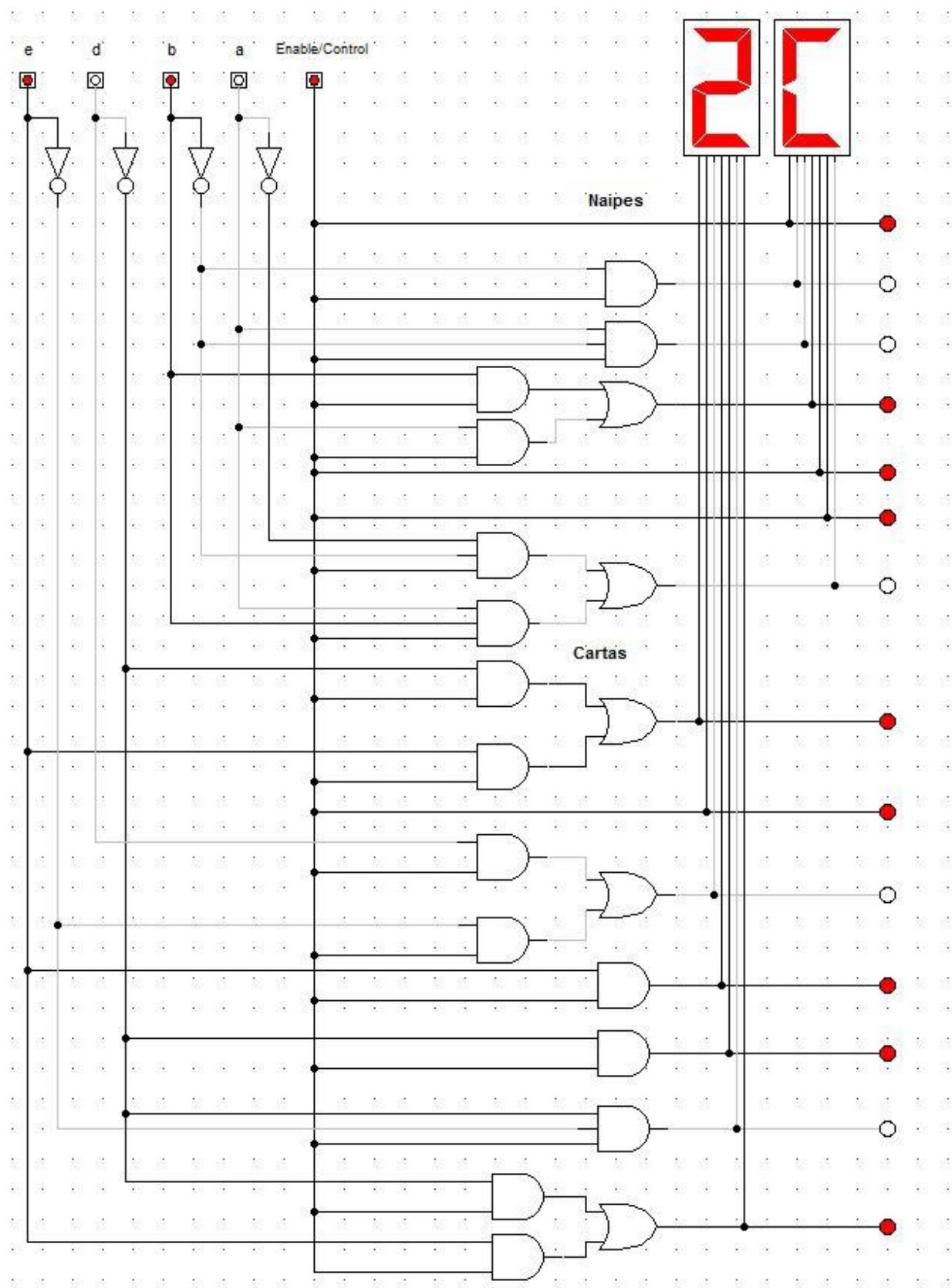


**Figura 5 – Alínea A – Circuito lógico – Com teste de entradas**

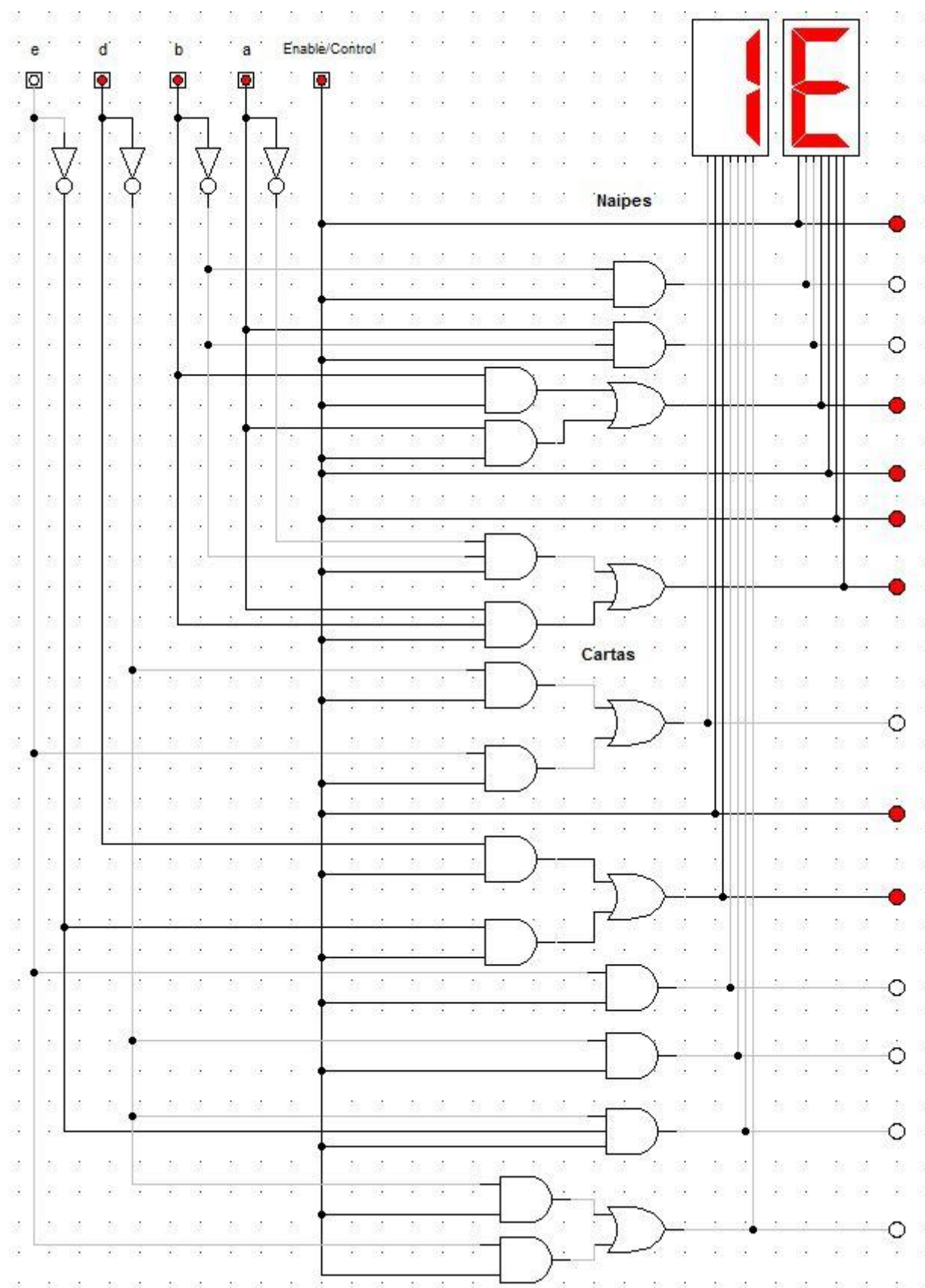




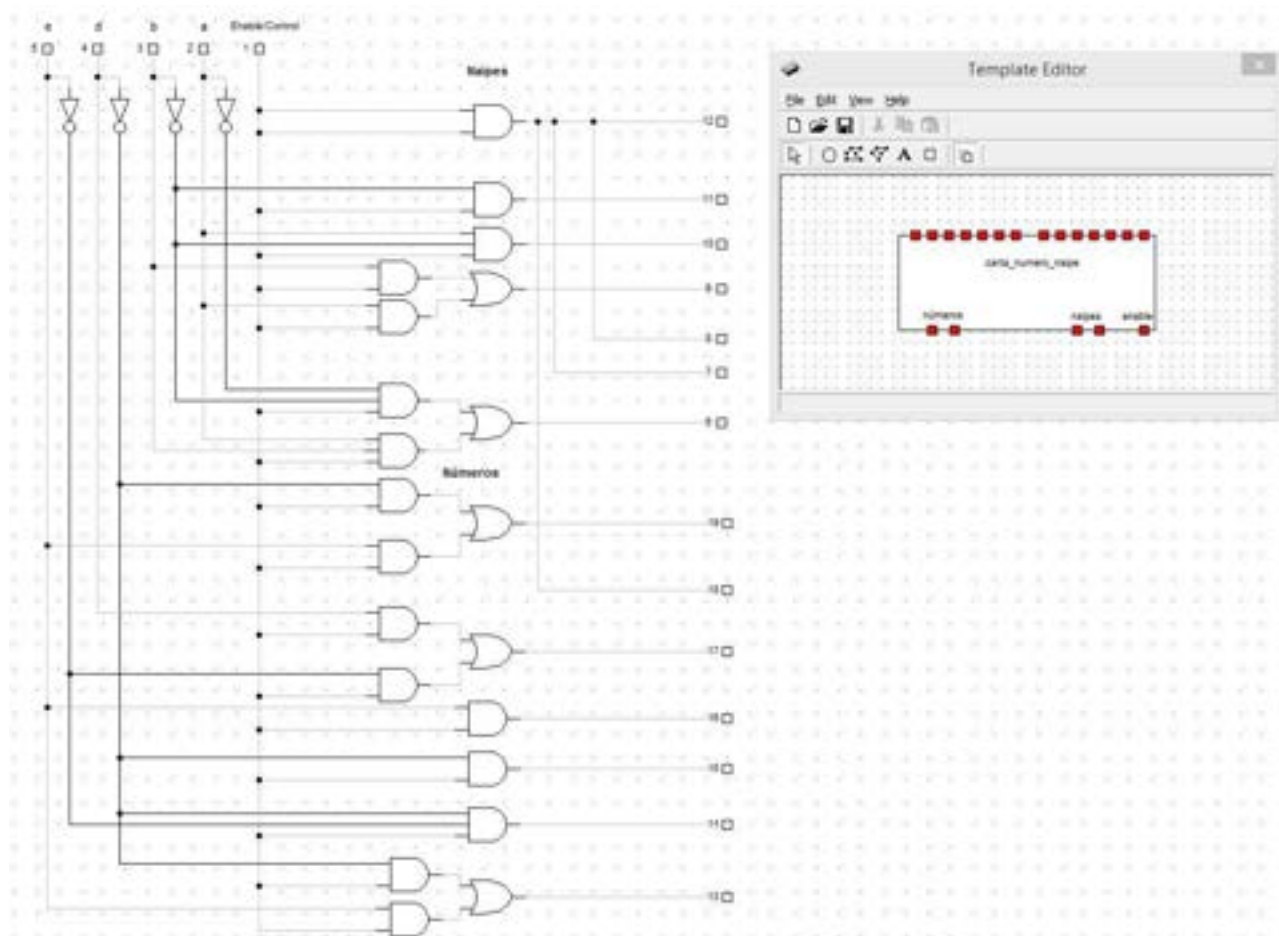
**Figura 6 – Alínea A – Circuito lógico – Com teste de entradas**



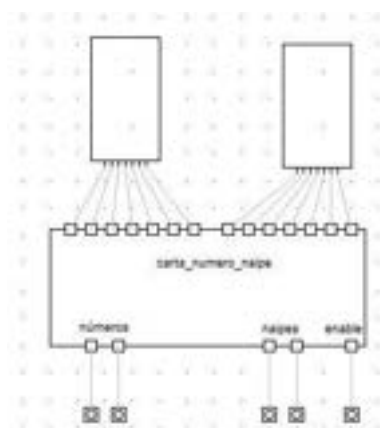
**Figura 7 – Alínea A – Circuito lógico – Com teste de entradas**



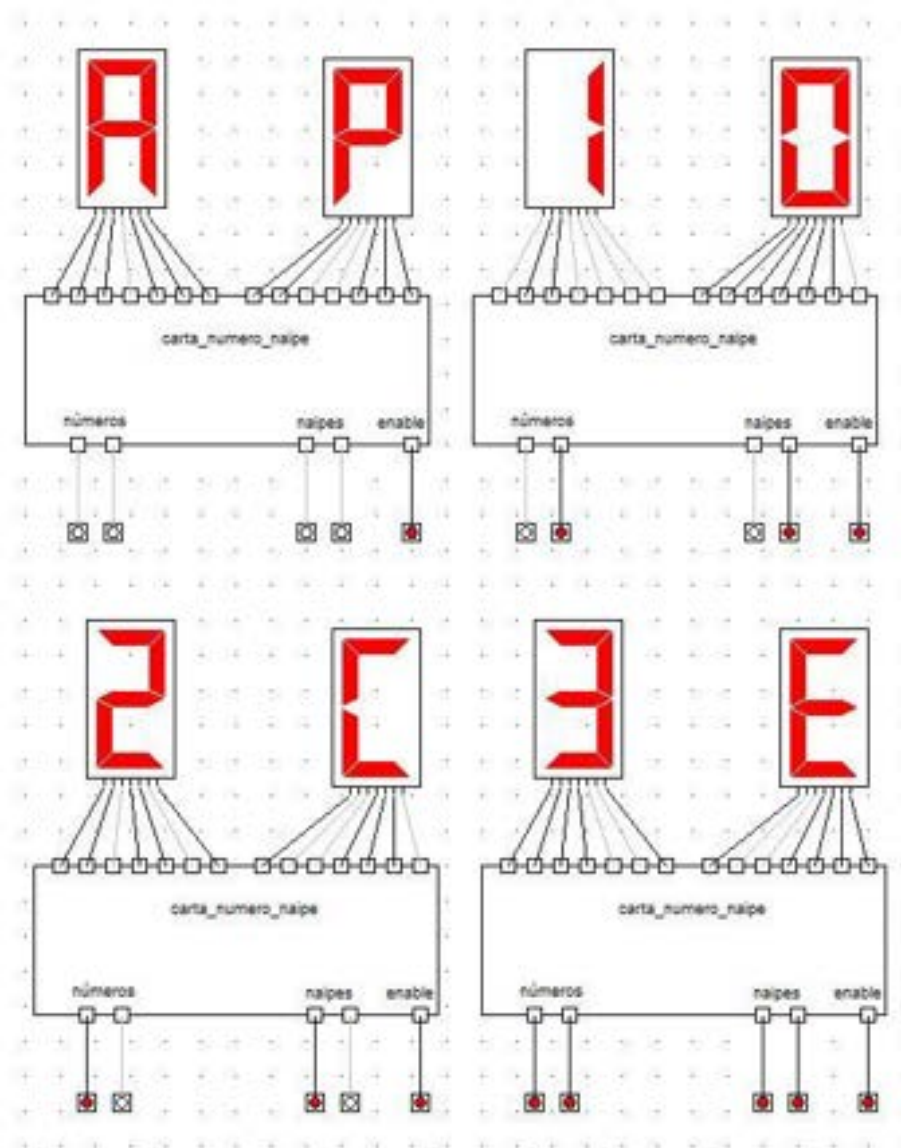
**Figura 8 – Alínea A – Circuito lógico – Com teste de entradas**



**Figura 9 – Alínea A – Criação do Componente**



**Figura 10 – Alínea A – Utilização do Componente – Com teste de entradas**



**Figura 11 – Alínea A – Utilização do Componente – Com testes de entradas**



Linhas				Colunas				Saída
L3	L2	L1	L0	C3	C2	C1	C0	
			1				1	0000
			1			1		0001
			1		1			0010
			1	1				0011
		1					1	0100
		1				1		0101
		1			1			0110
		1		1				0111
	1						1	1000
	1					1		1001
	1				1			1010
	1			1				1011
1							1	1100
1						1		1101
1					1			1110
1				1				1111

**Figura 12 – Alínea B – Mapeamento Linha/Coluna Saída**

Linhas				Colunas				Saída	s3	s2	s1	s0	s3	s2	s1	s0
L3	L2	L1	L0	C3	C2	C1	C0									
			1				1	0000	0	0	0	0	L2+L3	L1+L3	C2+C3	C1+C3
			1			1		0001	0	0	0	1	L2+L3	L1+L3	C2+C3	C1+C3
			1		1			0010	0	0	1	0	L2+L3	L1+L3	C2+C3	C1+C3
			1	1				0011	0	0	1	1	L2+L3	L1+L3	C2+C3	C1+C3
		1					1	0100	0	1	0	0	L2+L3	L1+L3	C2+C3	C1+C3
		1				1		0101	0	1	0	1	L2+L3	L1+L3	C2+C3	C1+C3
		1			1			0110	0	1	1	0	L2+L3	L1+L3	C2+C3	C1+C3
		1		1				0111	0	1	1	1	L2+L3	L1+L3	C2+C3	C1+C3
	1						1	1000	1	0	0	0	L2+L3	L1+L3	C2+C3	C1+C3
	1					1		1001	1	0	0	1	L2+L3	L1+L3	C2+C3	C1+C3
	1				1			1010	1	0	1	0	L2+L3	L1+L3	C2+C3	C1+C3
	1			1				1011	1	0	1	1	L2+L3	L1+L3	C2+C3	C1+C3
1							1	1100	1	1	0	0	L2+L3	L1+L3	C2+C3	C1+C3
1						1		1101	1	1	0	1	L2+L3	L1+L3	C2+C3	C1+C3
1					1			1110	1	1	1	0	L2+L3	L1+L3	C2+C3	C1+C3
1				1				1111	1	1	1	1	L2+L3	L1+L3	C2+C3	C1+C3

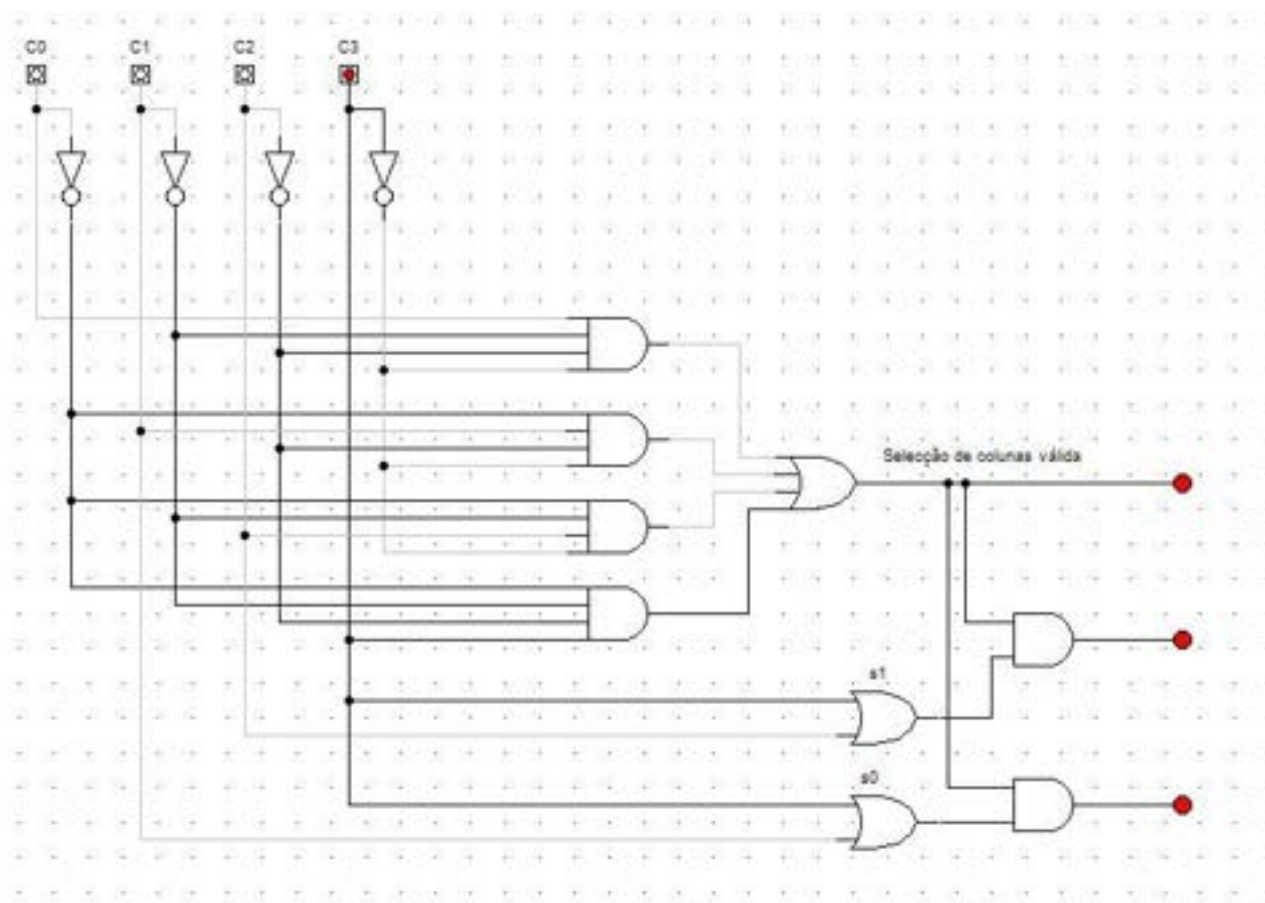
**Figura 13 – Alínea B – Expressões lógicas das saídas**

Linhas				Linhas	Colunas				Colunas
L3	L2	L1	L0	Enable	C3	C2	C1	C0	Enable
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	1	1
0	0	1	0	1	0	0	1	0	1
0	0	1	1	0	0	0	1	1	0
0	1	0	0	1	0	1	0	0	1
0	1	0	1	0	0	1	0	1	0
0	1	1	0	0	0	1	1	0	0
0	1	1	1	0	0	1	1	1	0
1	0	0	0	1	1	0	0	0	1
1	0	0	1	0	1	0	0	1	0
1	0	1	0	0	1	0	1	0	0
1	0	1	1	0	1	0	1	1	0
1	1	0	0	0	1	1	0	0	0
1	1	0	1	0	1	1	0	1	0
1	1	1	1	0	1	1	1	1	0

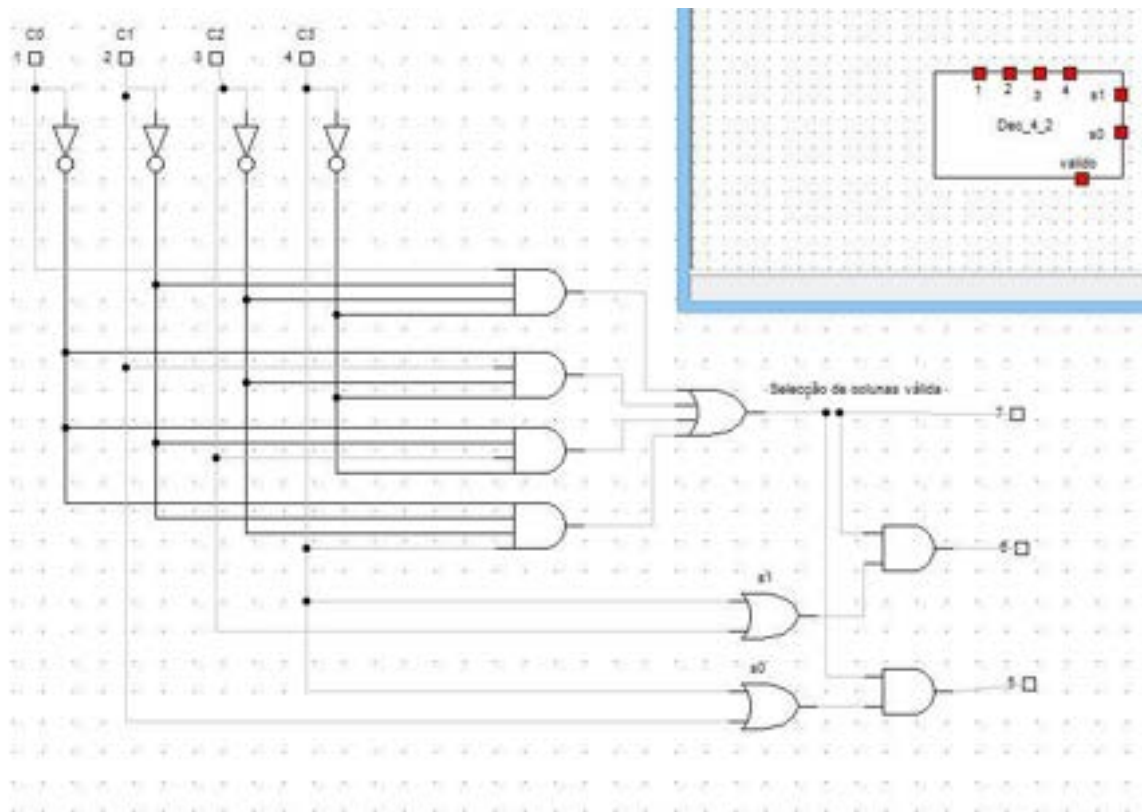
  

Colunas	Linhas
C0/C1/C2/C3	L0/L1/L2/L3
/C0C1/C2/C3	/L0L1/L2/L3
/C0/C1C2/C3	/L0/L1L2/L3
/C0/C1/C2C3	/L0/L1/L2L3

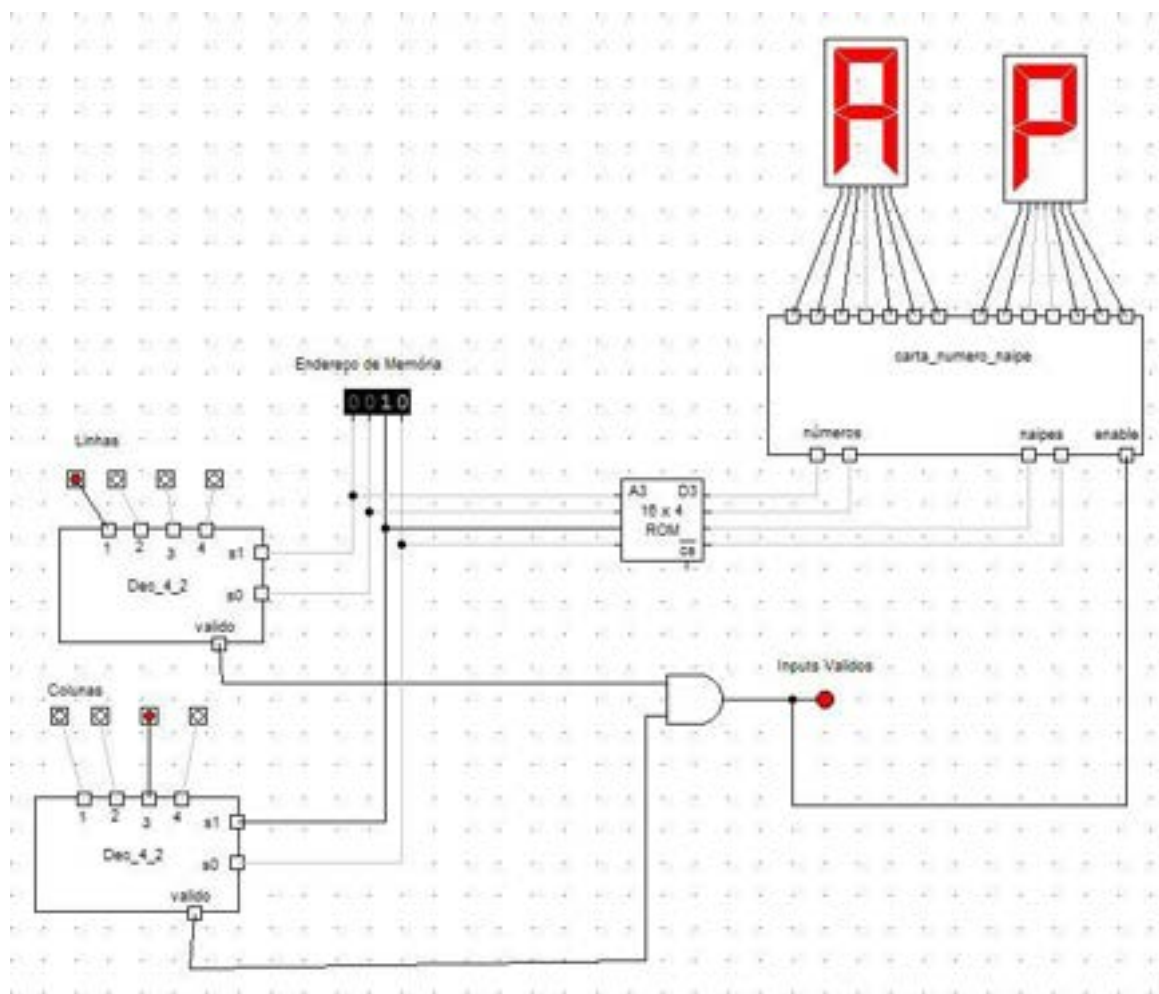
**Figura 14 – Alínea B – Cálculo do enable ou de selecção válida das linhas/colunas**



**Figura 15 – Alínea B – Circuito lógico**

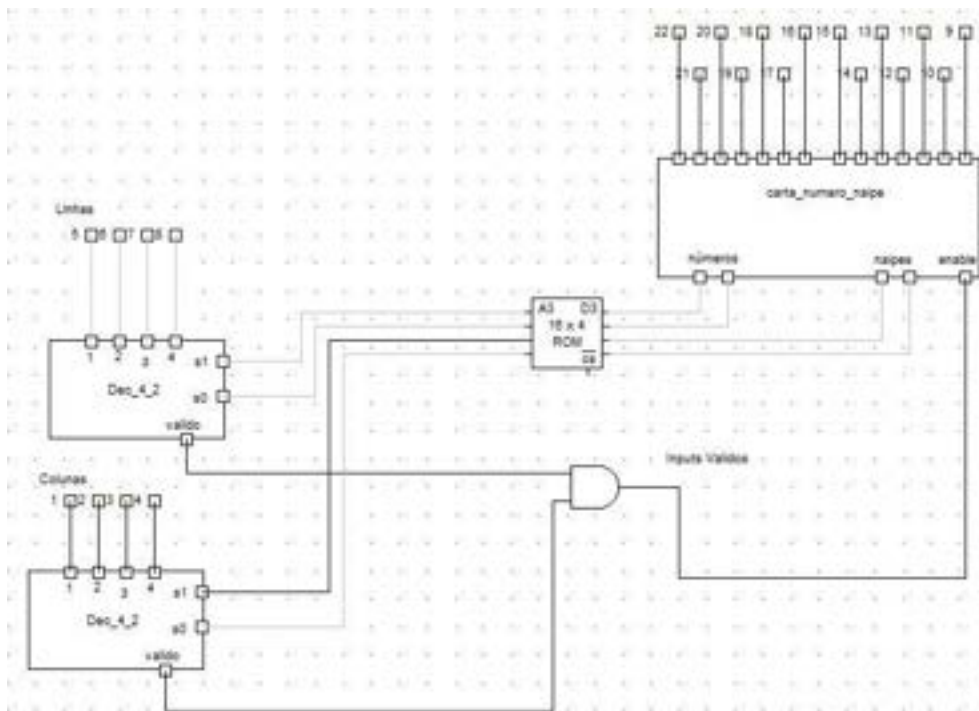


**Figura 16 – Alínea B – Componente com duas entradas e duas saídas**

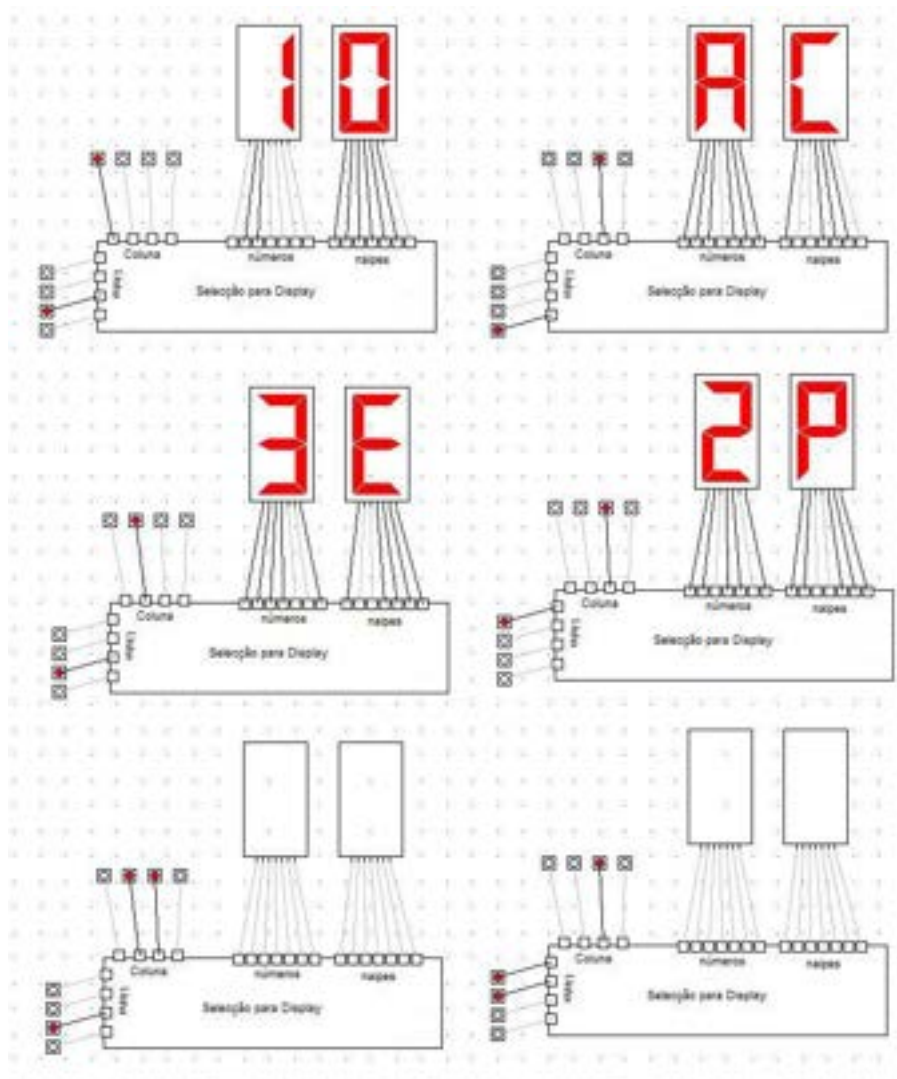


**Figura 17 – Alínea B – Circuito lógico utilizando 2 componentes**

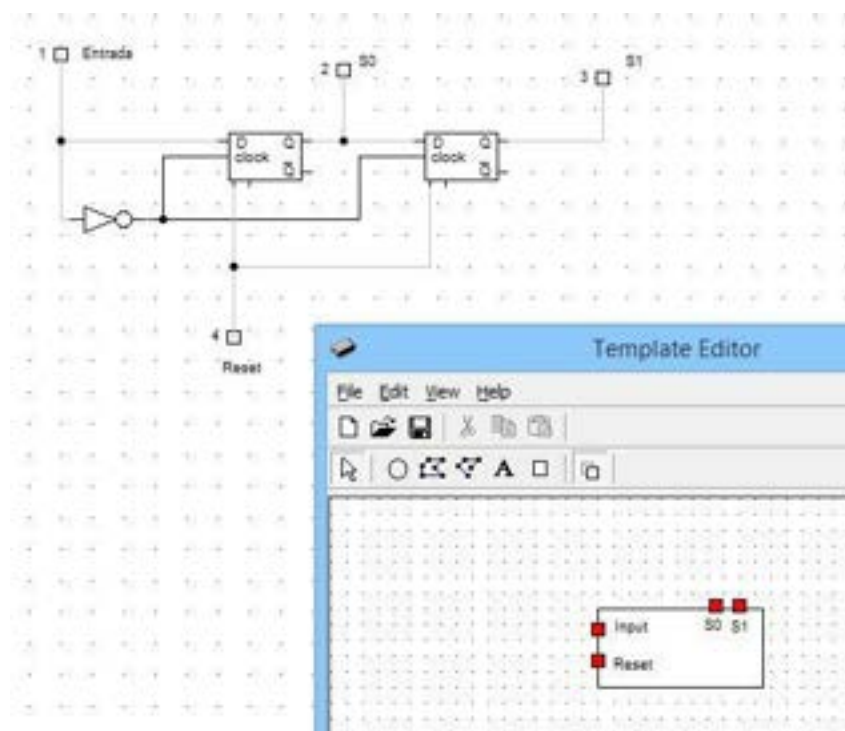




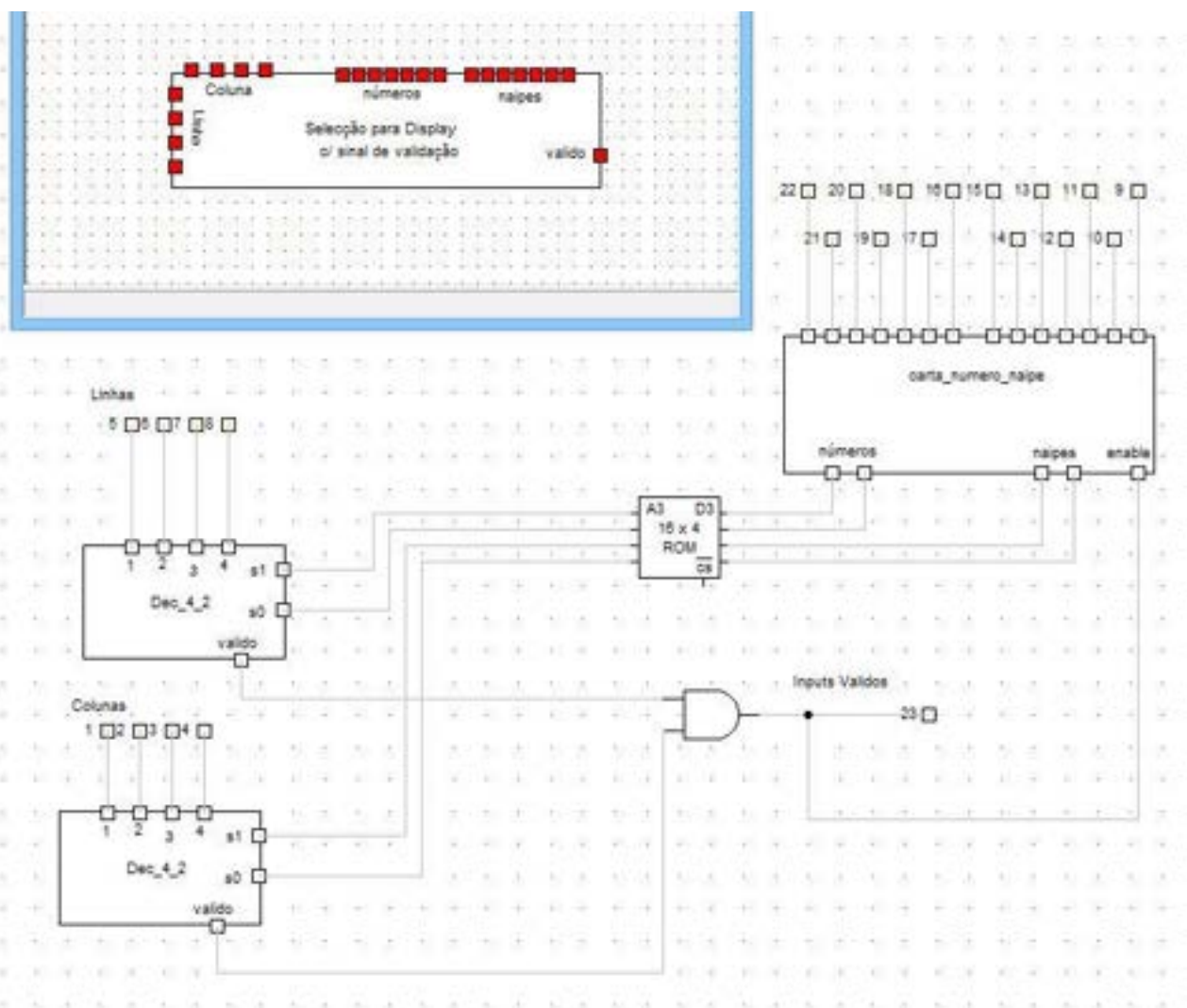
**Figura 18 – Alínea B – Componente final**



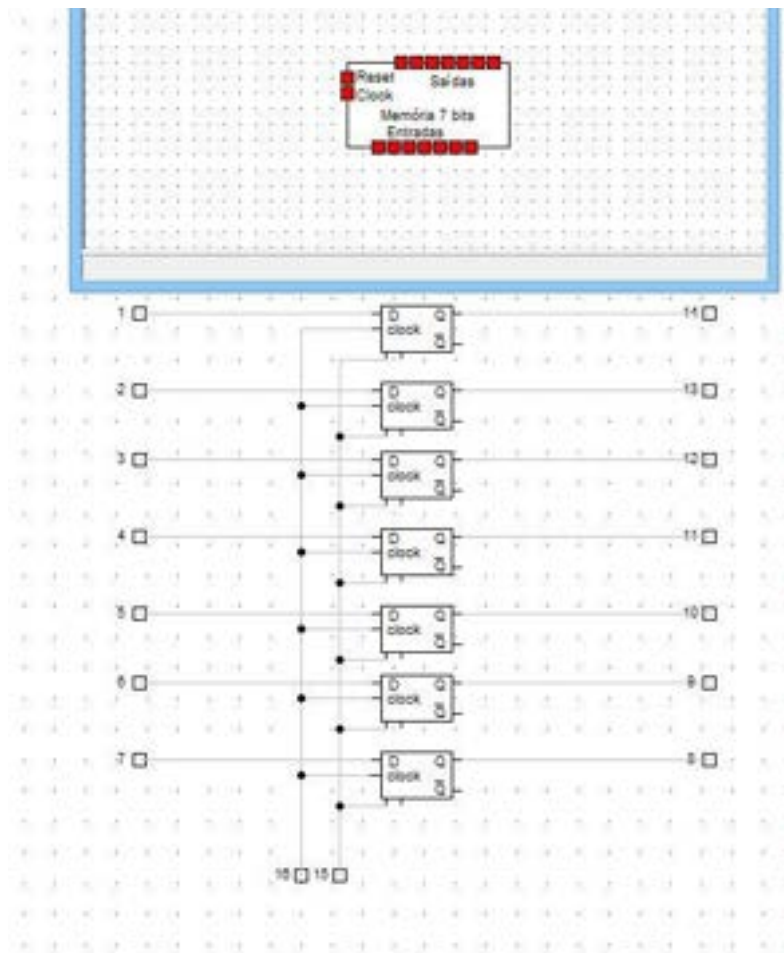
**Figura 19 – Alínea B – Utilização do Componente – Com testes de entradas**



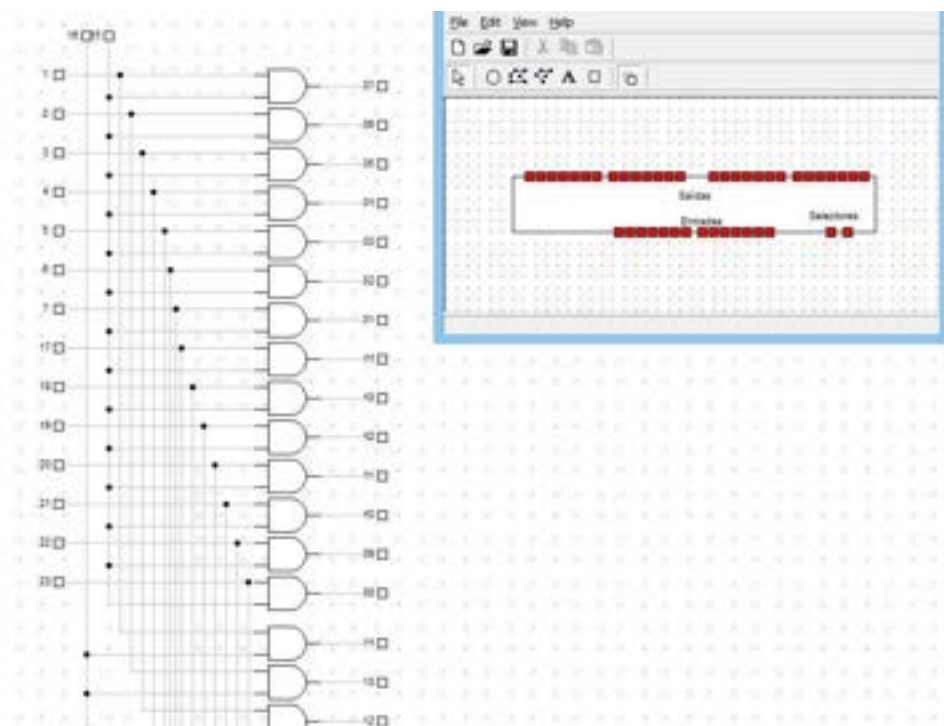
**Figura 20 – Alínea C – Componente contador**



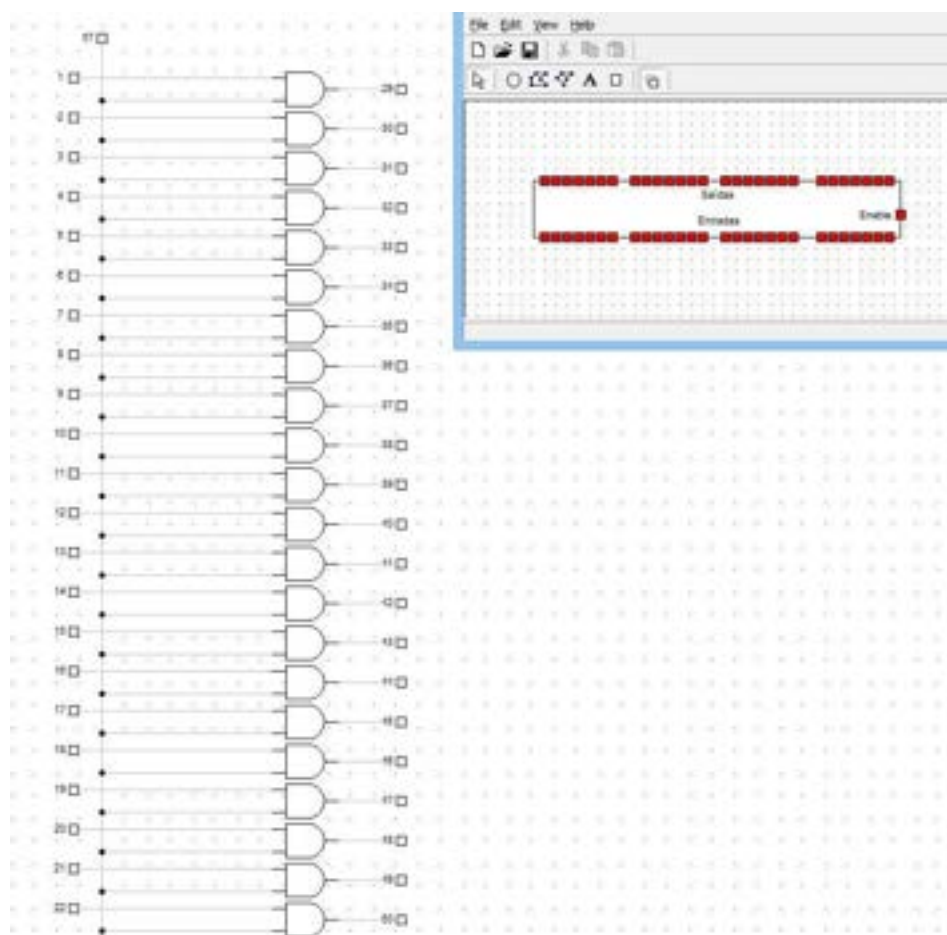
**Figura 21 – Alínea C – Componente Aline B com output de entradas válidas**



**Figura 22 – Alínea C – Componente Memória 7 bits**



**Figura 23 – Alínea C – Componente Multiplexador de Seleção**



**Figura 24 – Alínea C – Componente Buffer**

#### Alínea C

	Contador ( S0 S1)			Selector de Memória		Buffer	
Passo 1	00	0	0	A	$/(S0+S1)$	0	
Passo 2	01	0	1			0	
Passo 3	10	1	0	B	$S0/S1$	0	
Passo 4	11	1	1			1	S0S1
Passo 5	11	1	1			1	

#### Reset contador

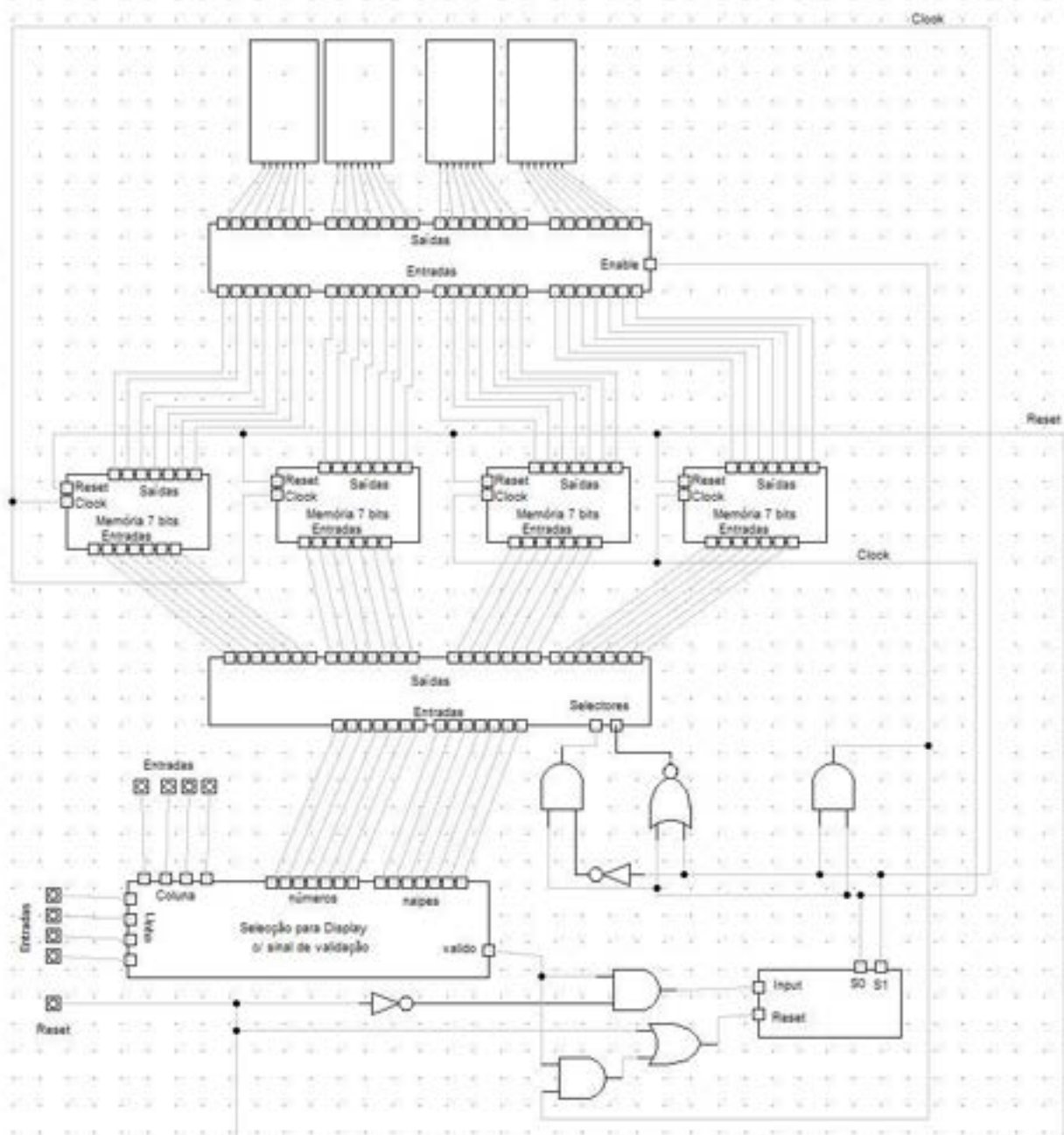
Quando  $S0$  and  $S1 = 1$ , ou seja, Buffer ativo e entradas válidas

Quando ligado na entrada

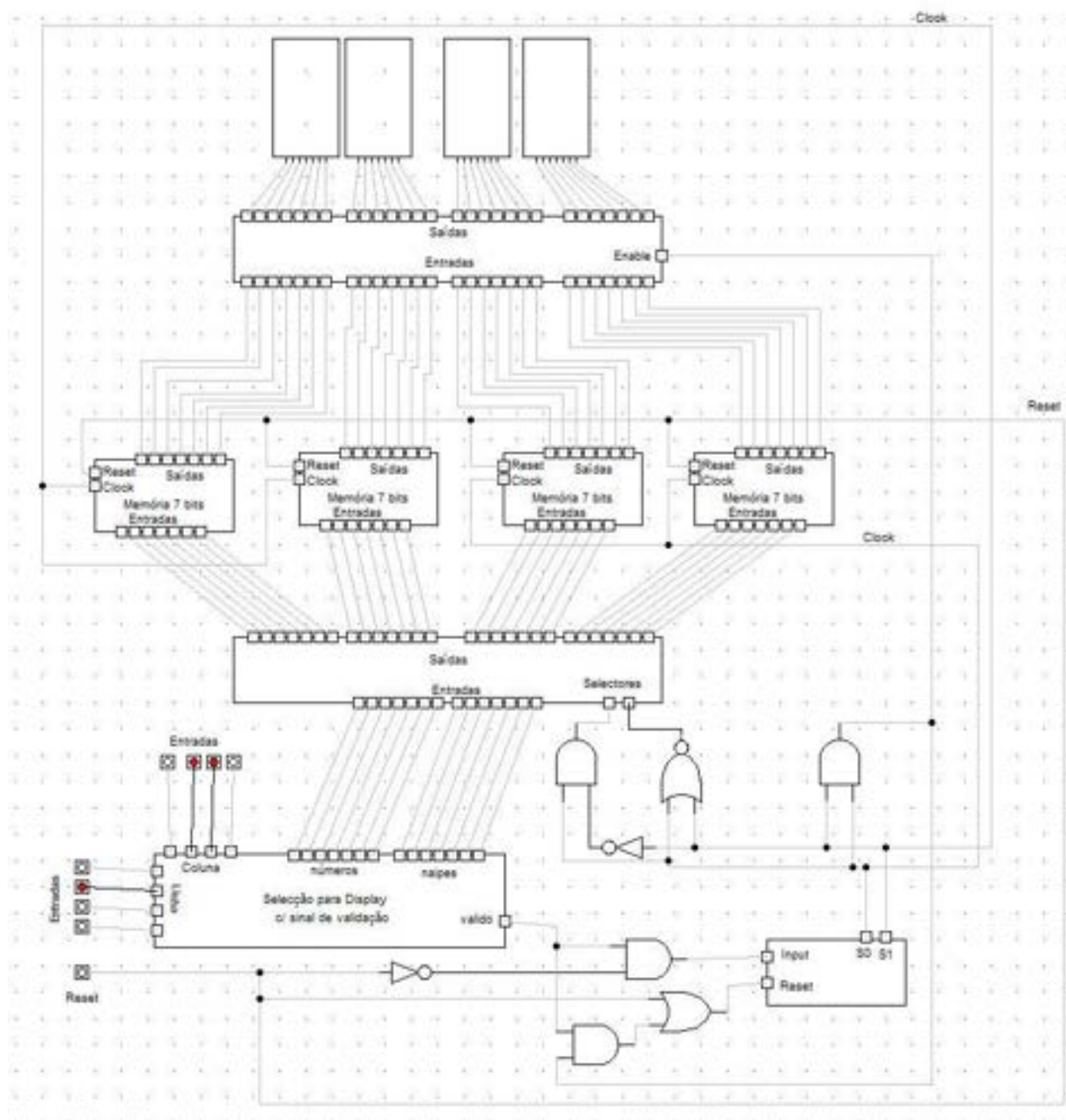
Contador só conta com reset desligado

**Figura 25 – Alínea C – Estudo e validações**

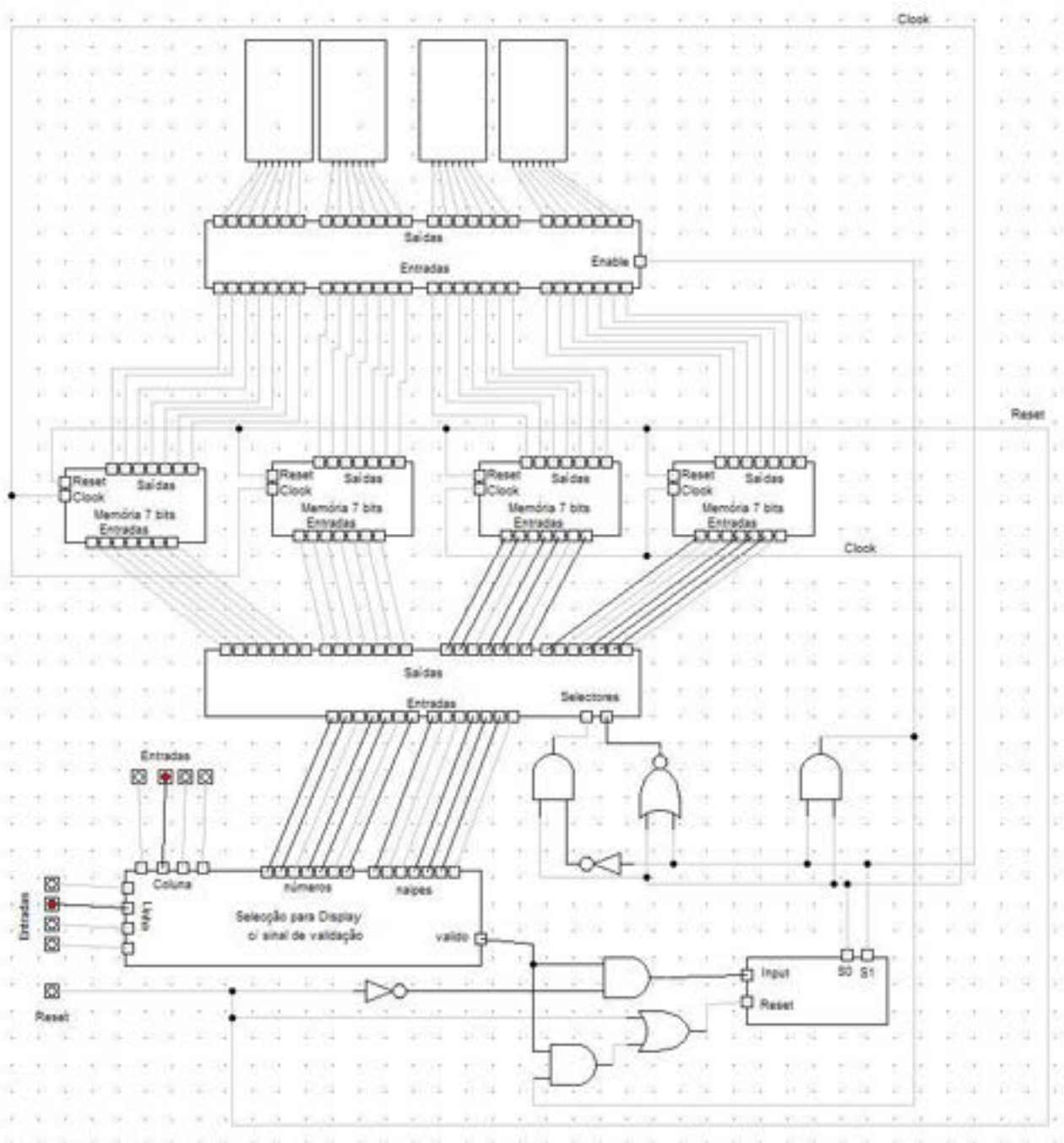




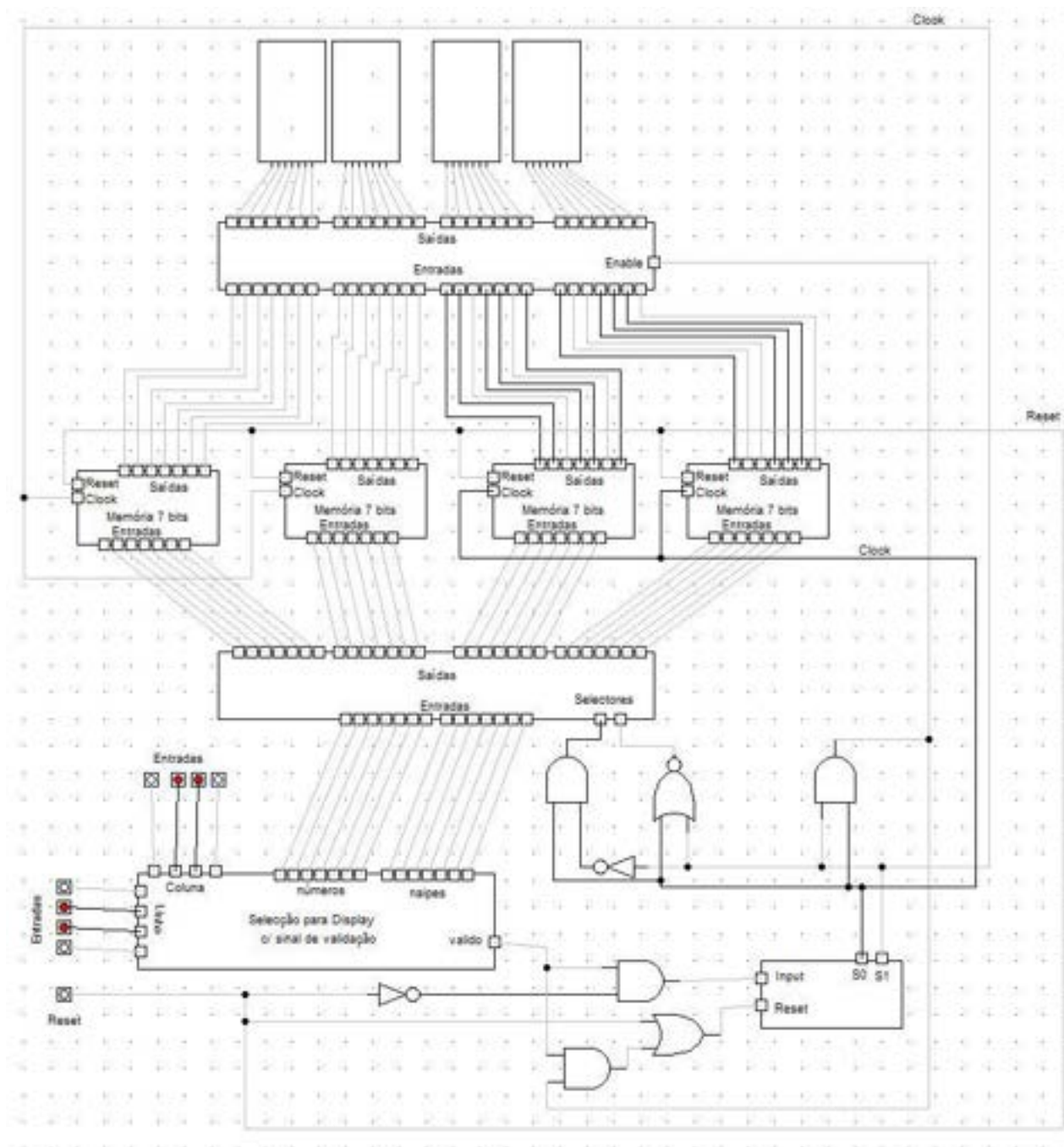
**Figura 26 – Alínea C – Circuito**



**Figura 27 – Alínea C – Circuito com testes – Passo 1**

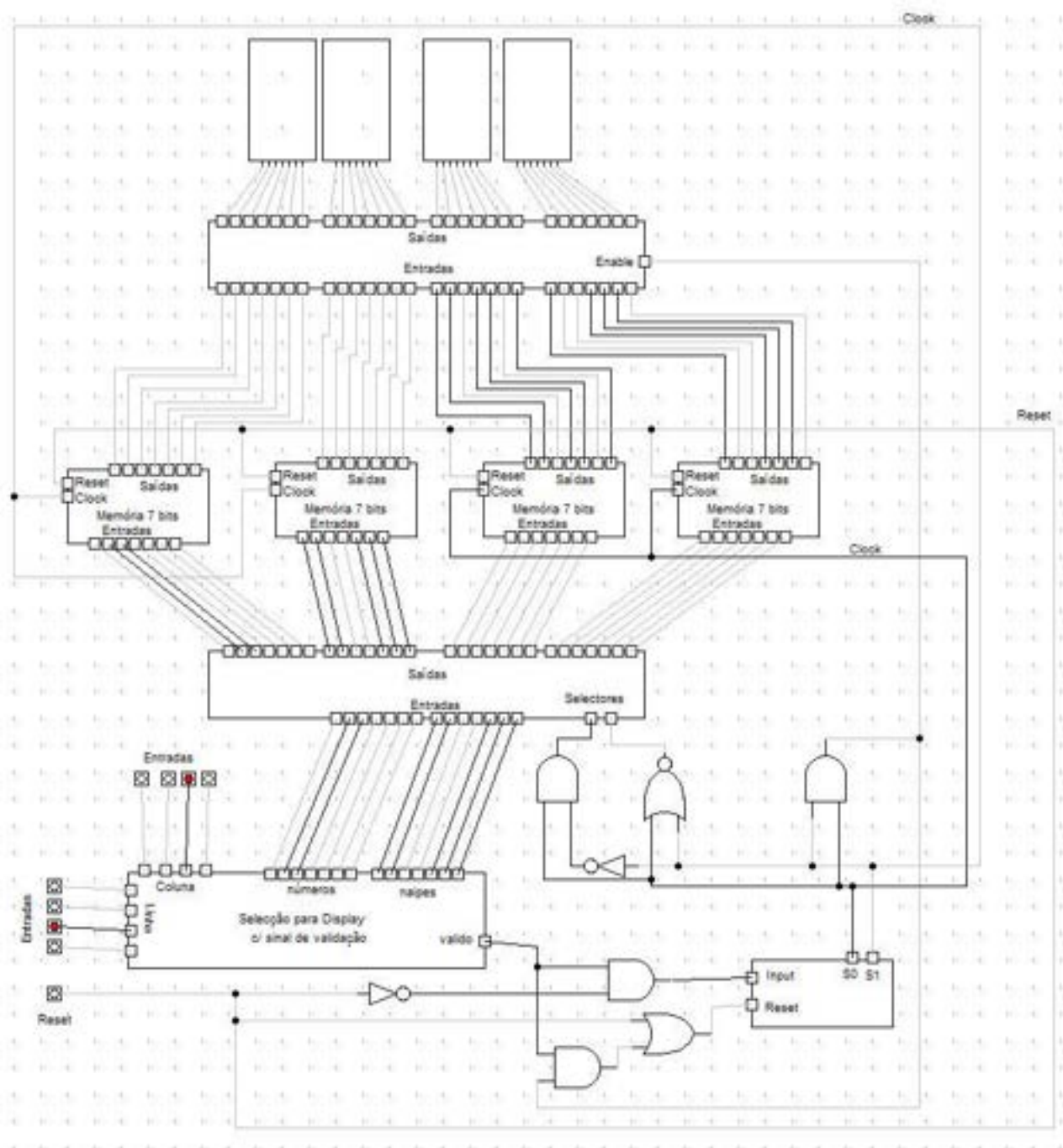


**Figura 28 – Alínea C – Circuito com testes – Passo 2**

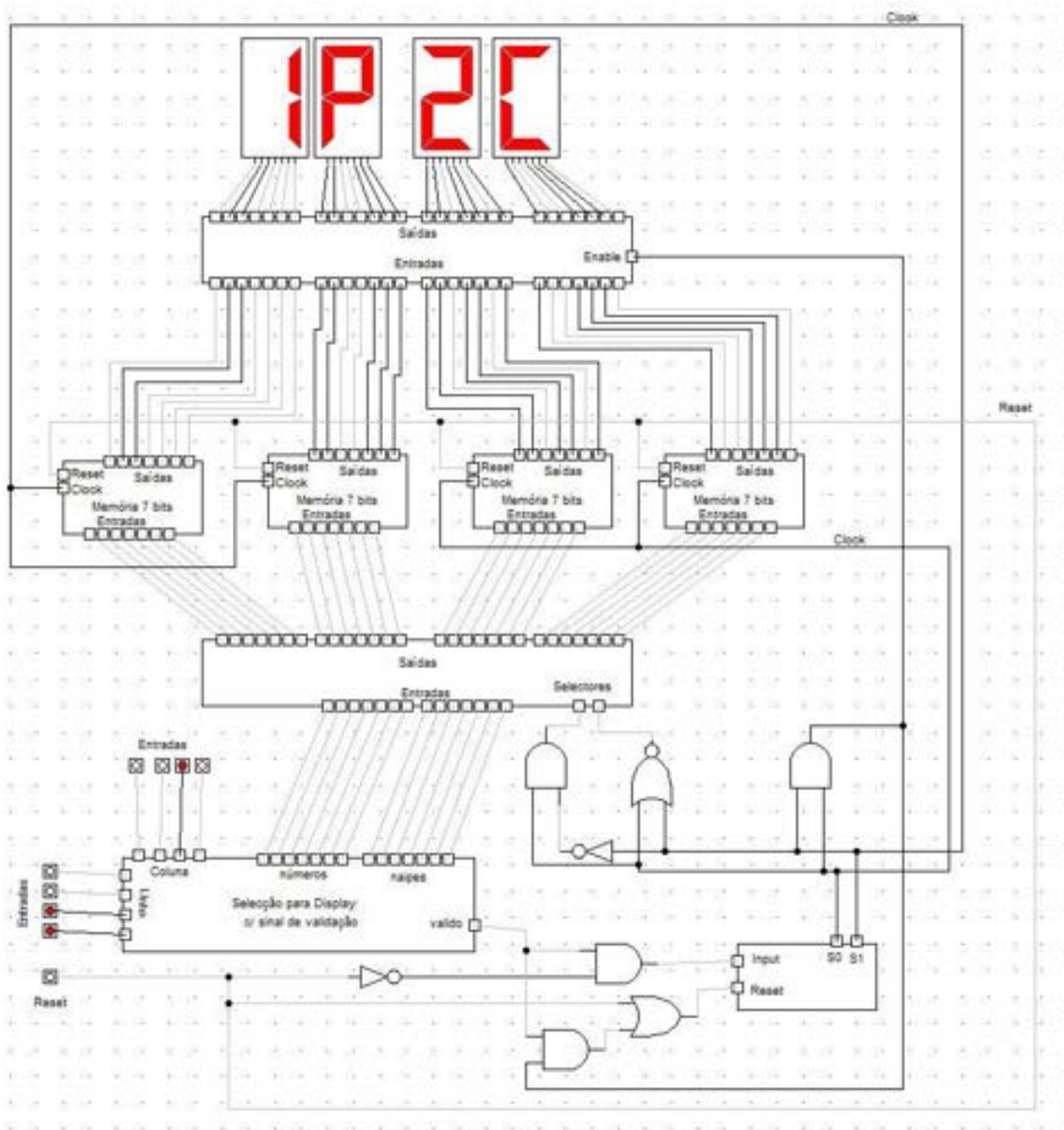


**Figura 29 – Alínea C – Circuito com testes – Passo 3**

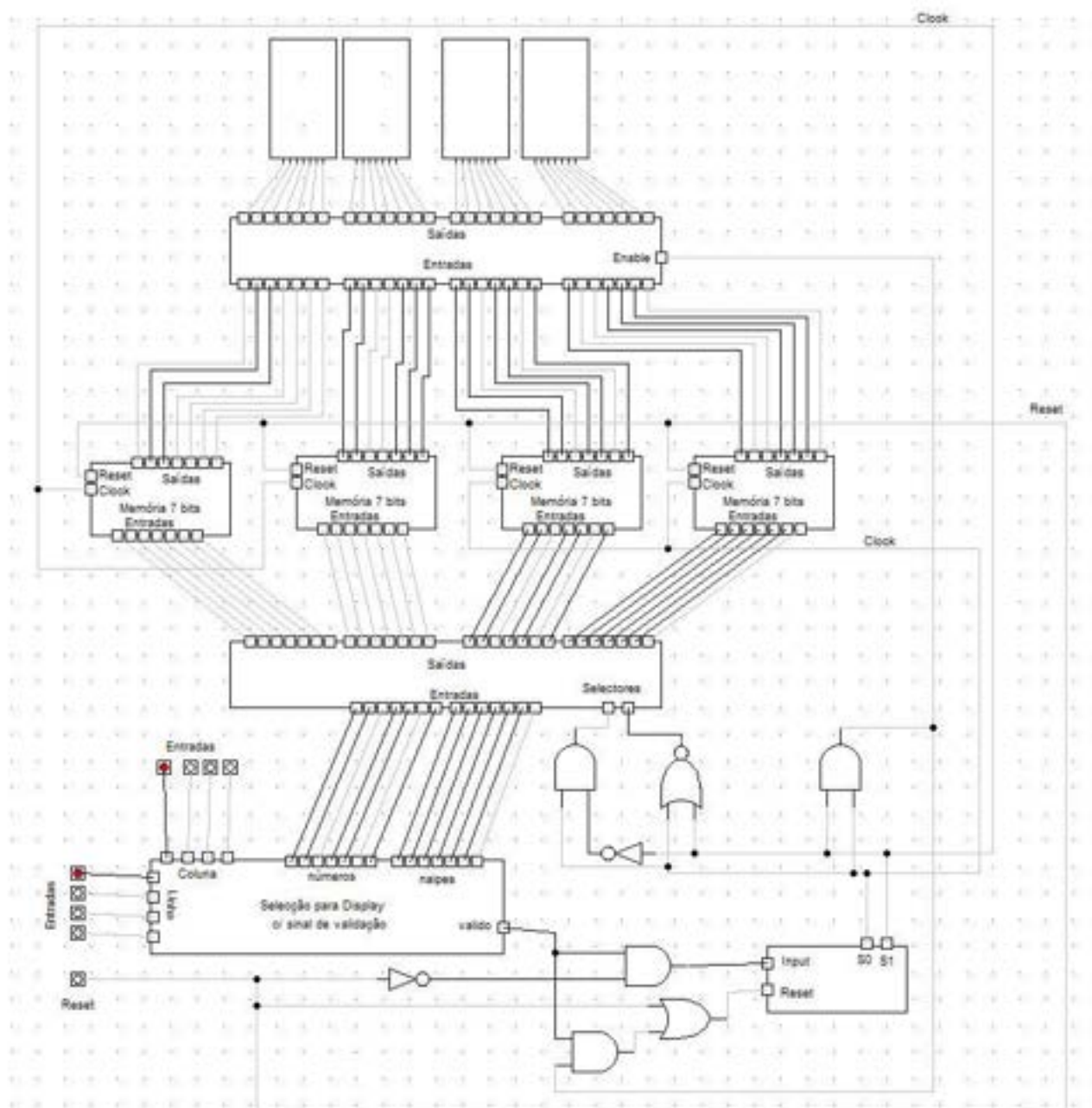




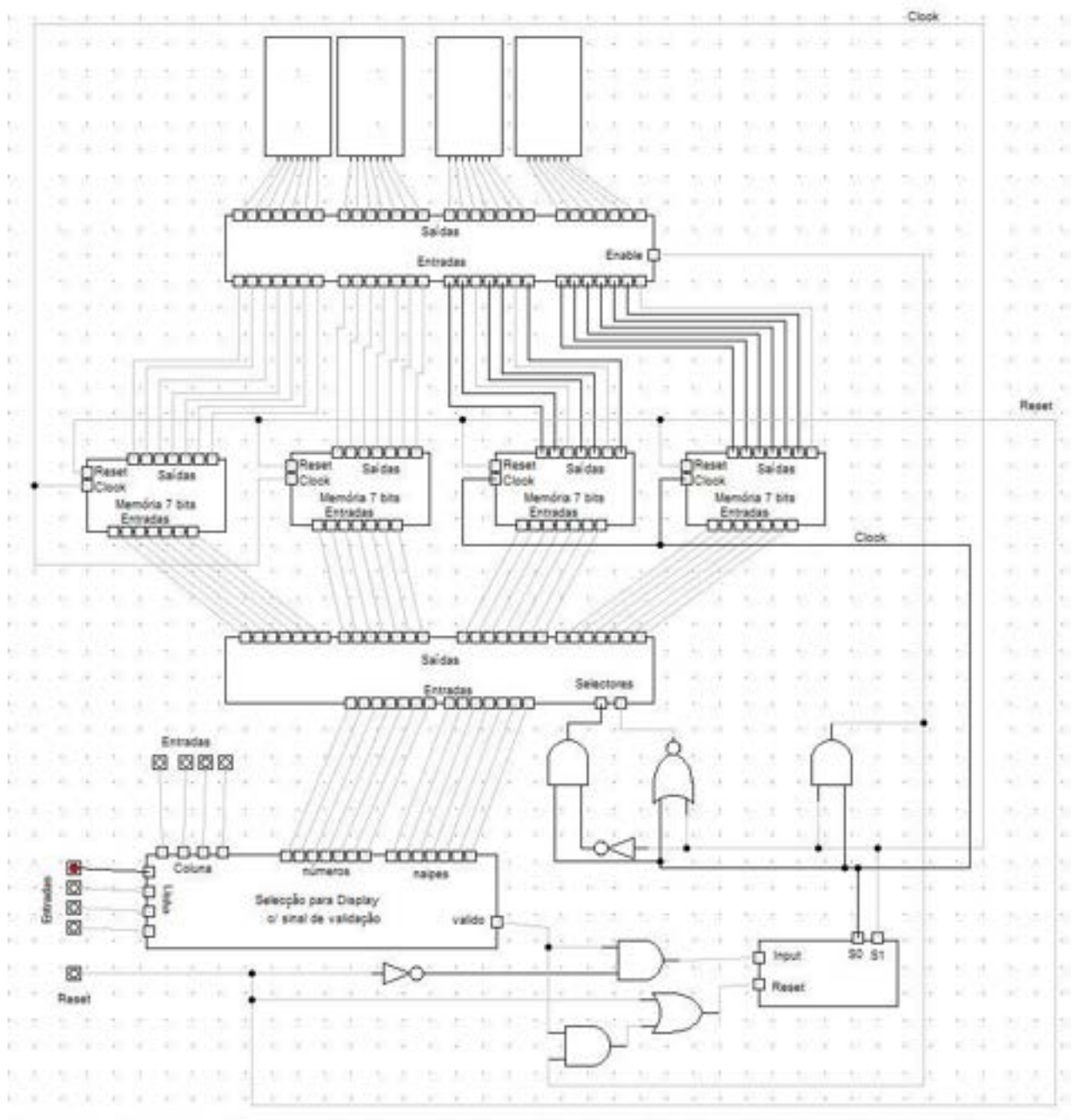
**Figura 30 – Alínea C – Circuito com testes – Passo 4**



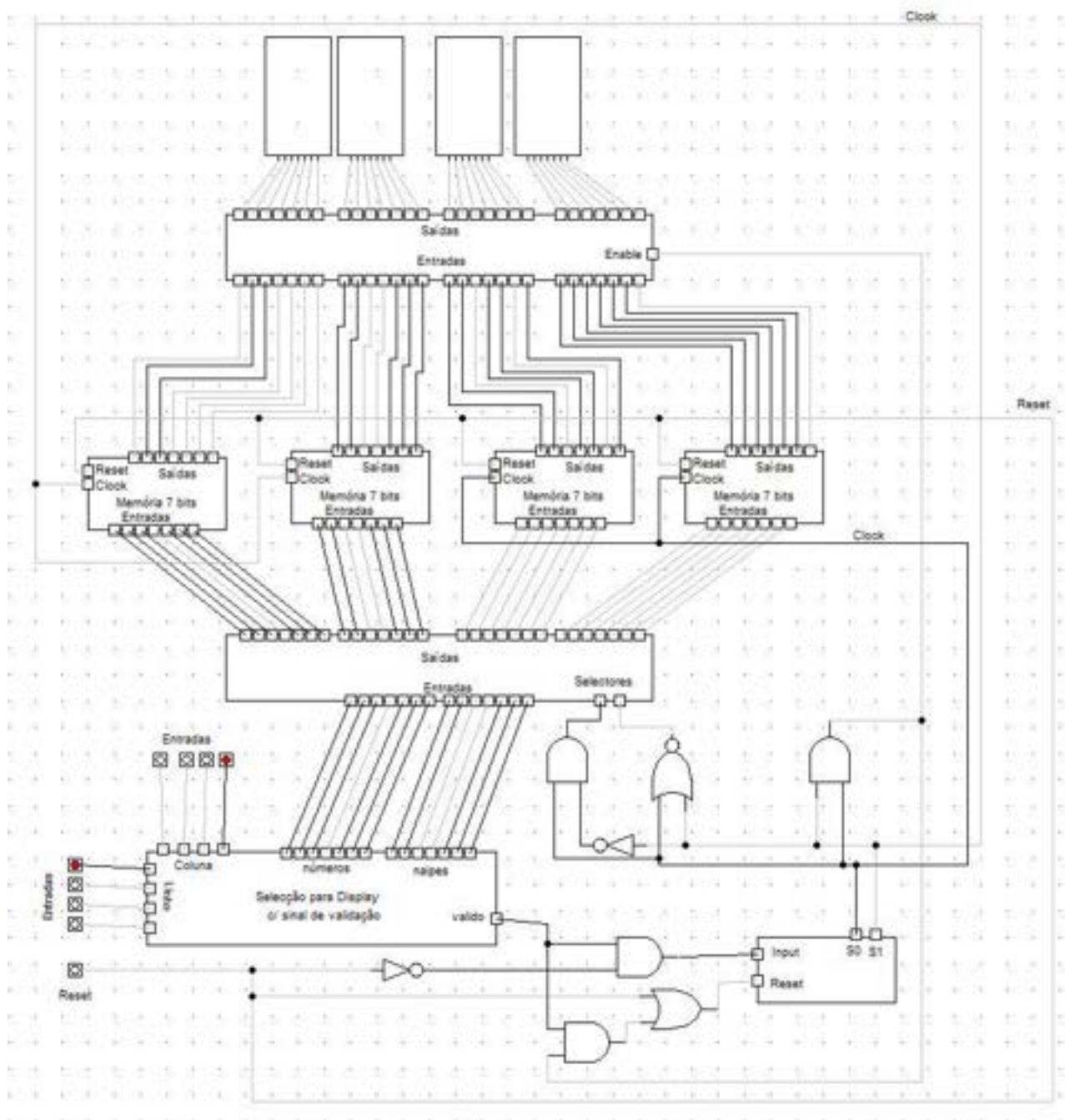
**Figura 31 – Alínea C – Circuito com testes – Passo 5**



**Figura 32 – Alínea C – Circuito com testes – Passo 2 (novamente)**

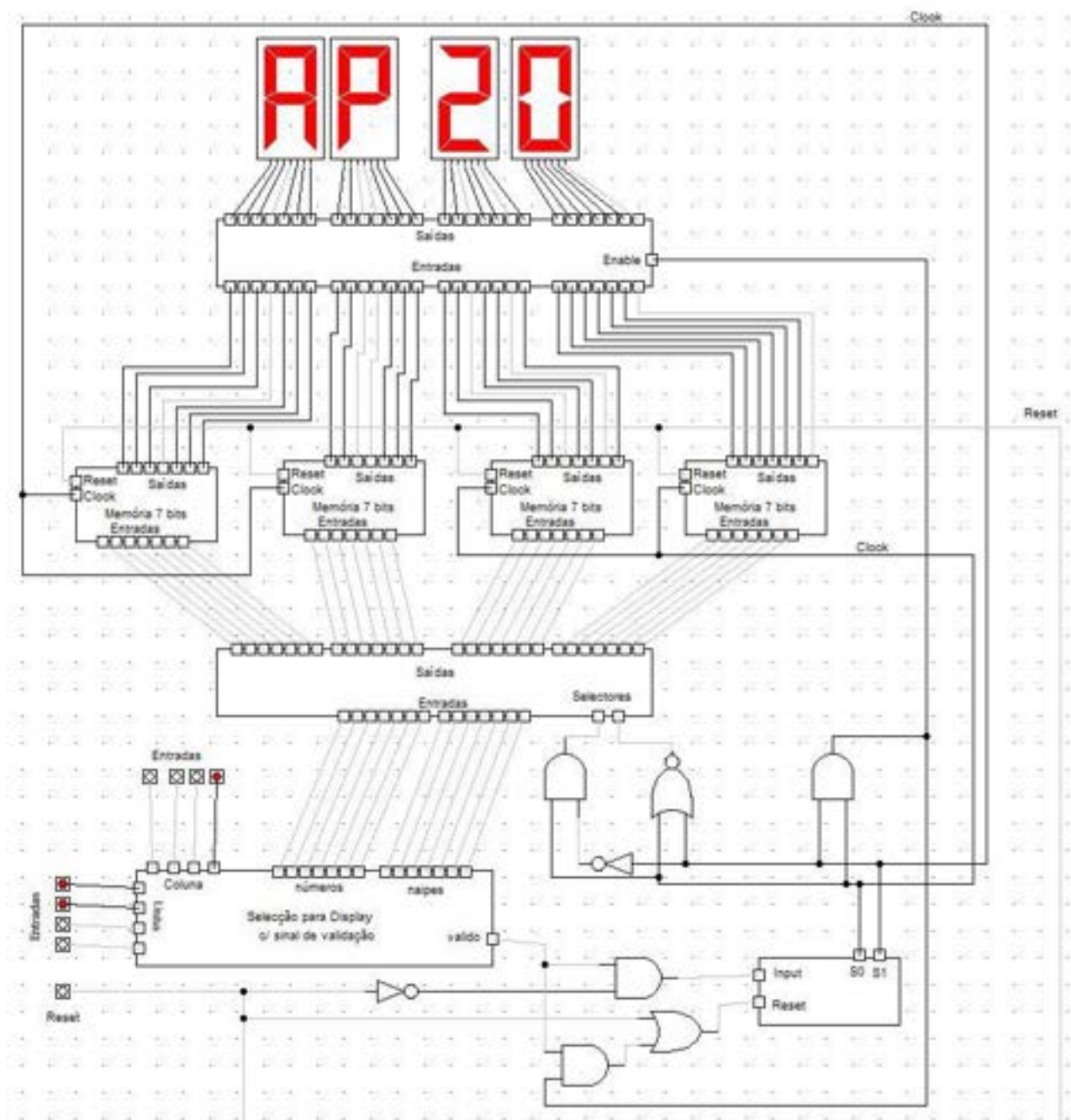


**Figura 33 – Alínea C – Circuito com testes – Passo 3**

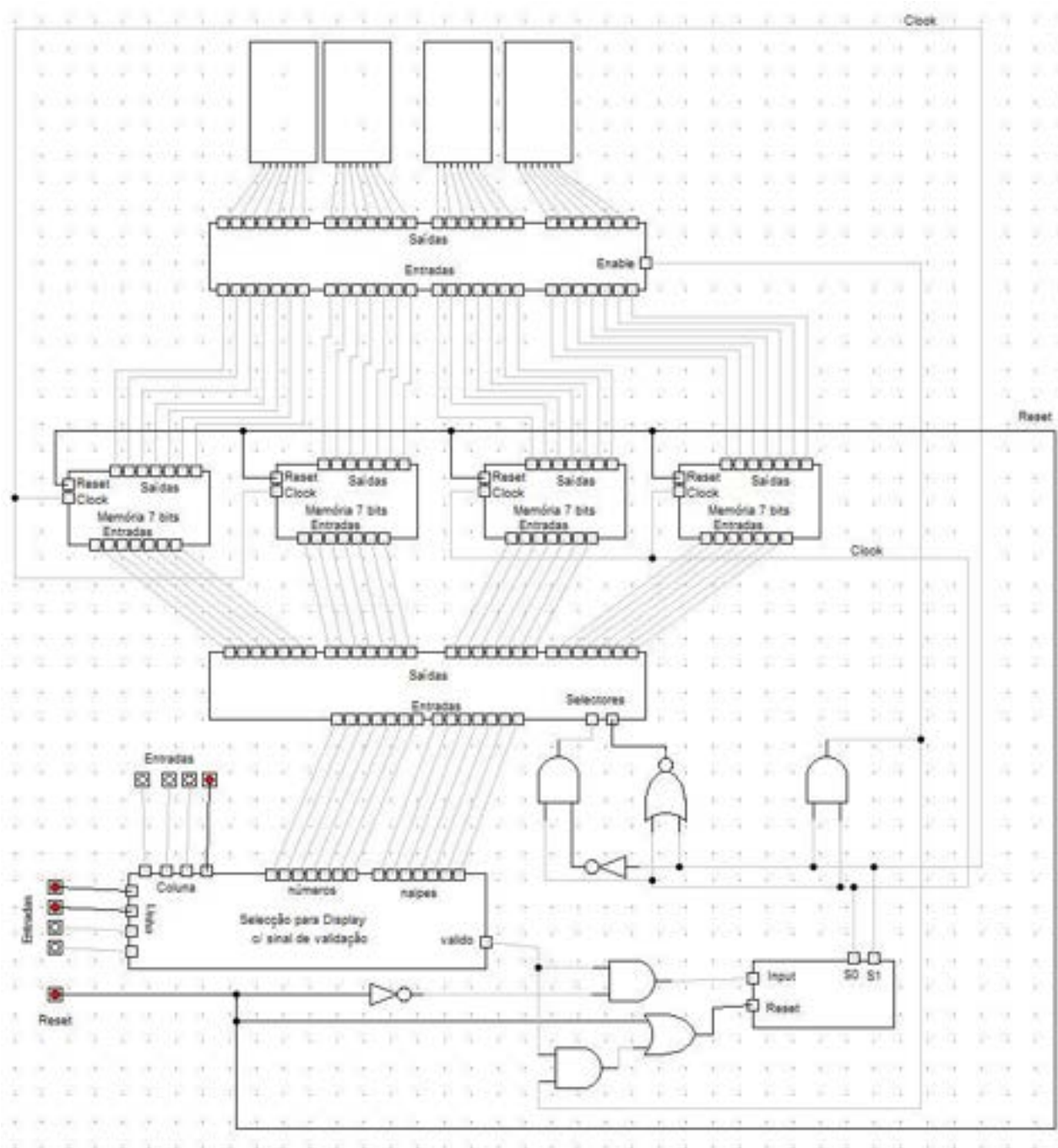


**Figura 34 – Alínea C – Circuito com testes – Passo 4**

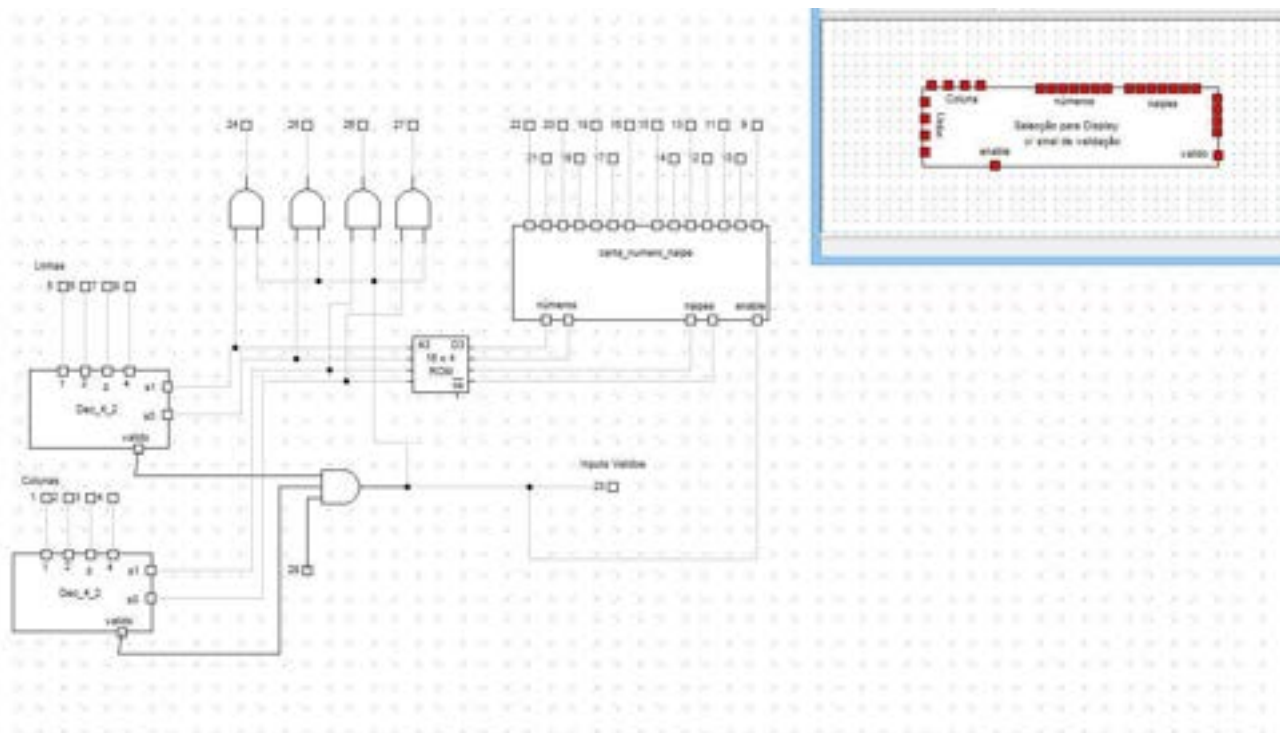




**Figura 35 – Alínea C – Circuito com testes – Passo 5**

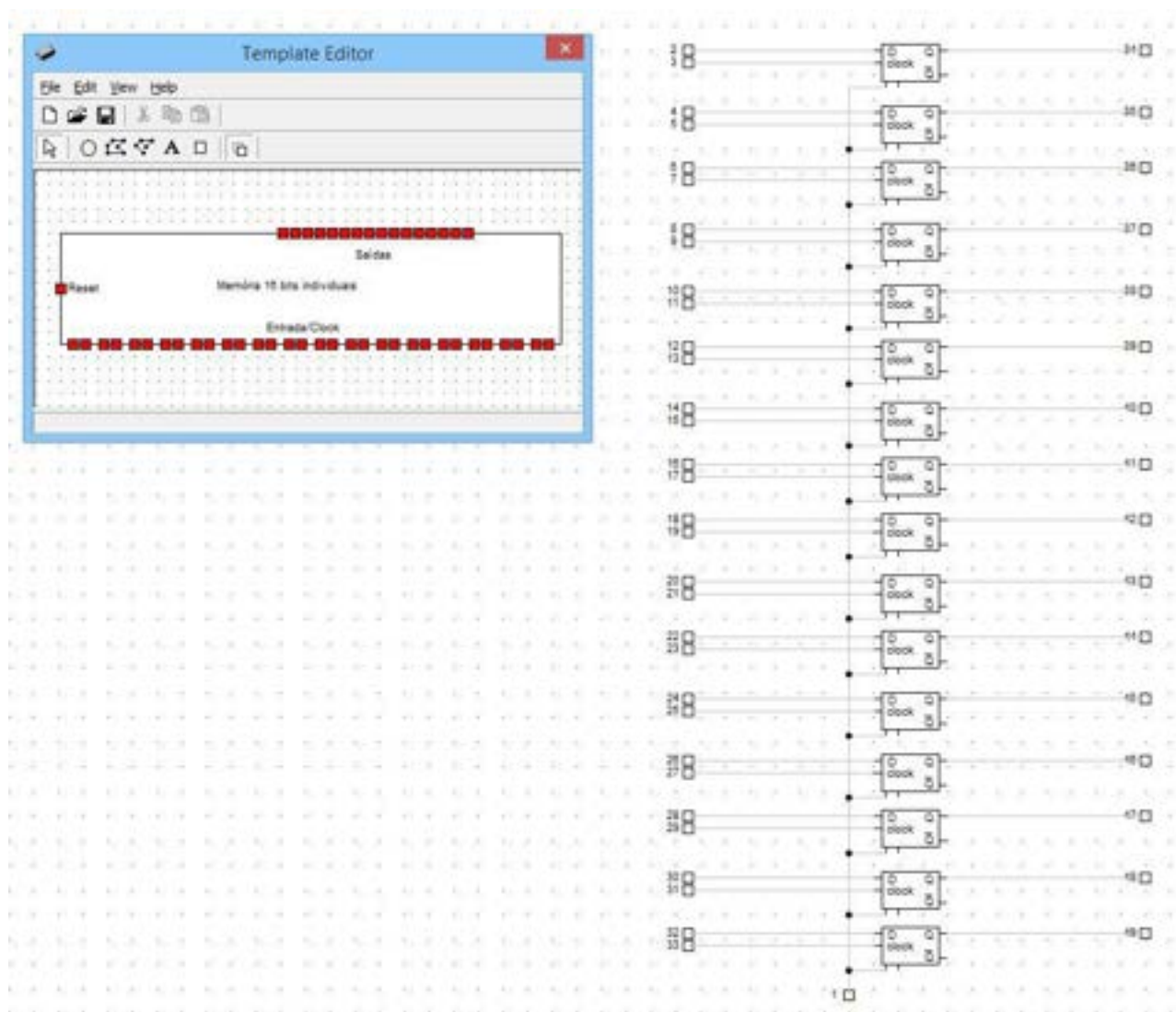


**Figura 36 – Alínea C – Circuito com testes – Reset**

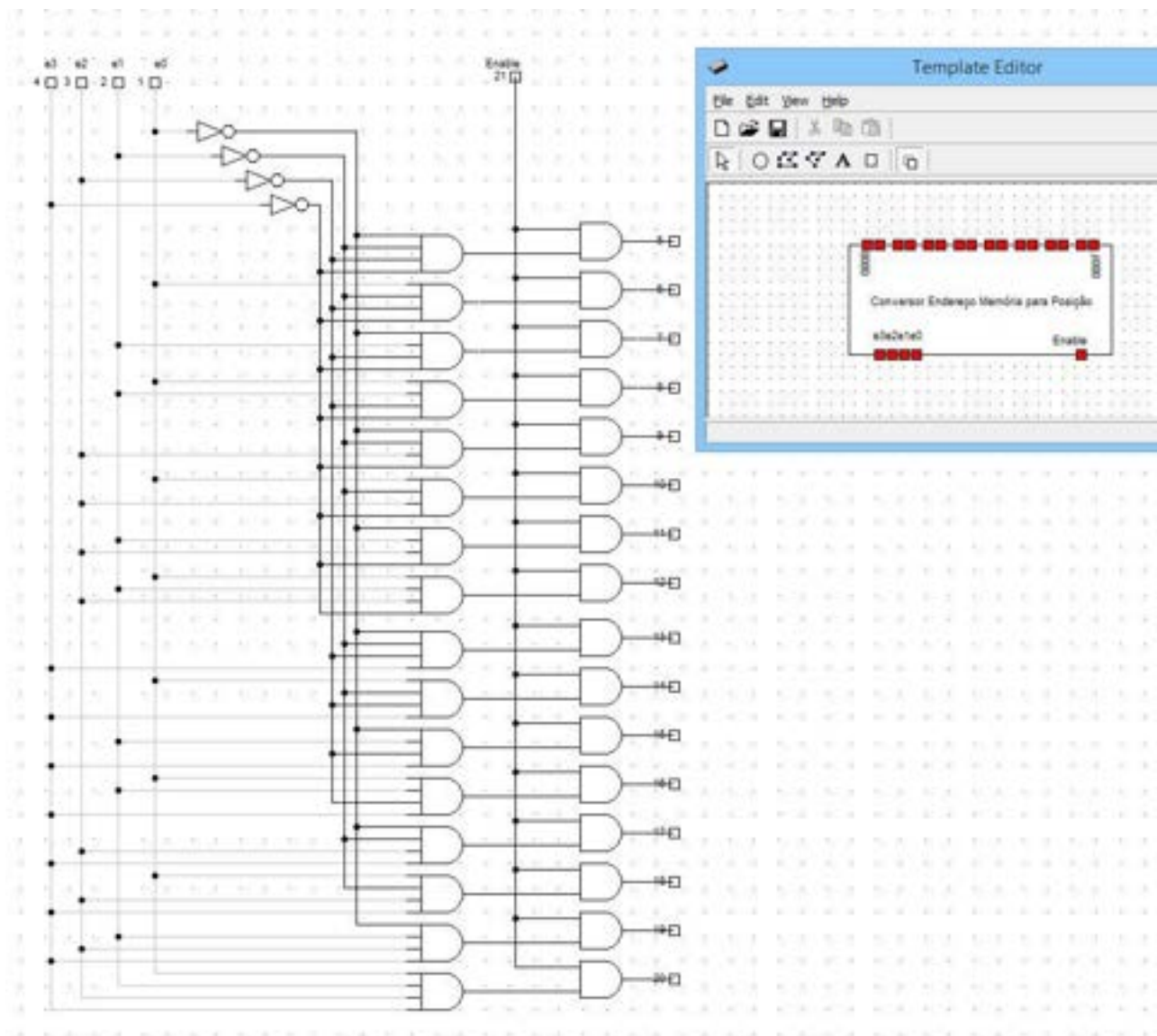


**Figura 37 – Alínea D – Componente com output de entradas válidas, endereço de memória e enable**

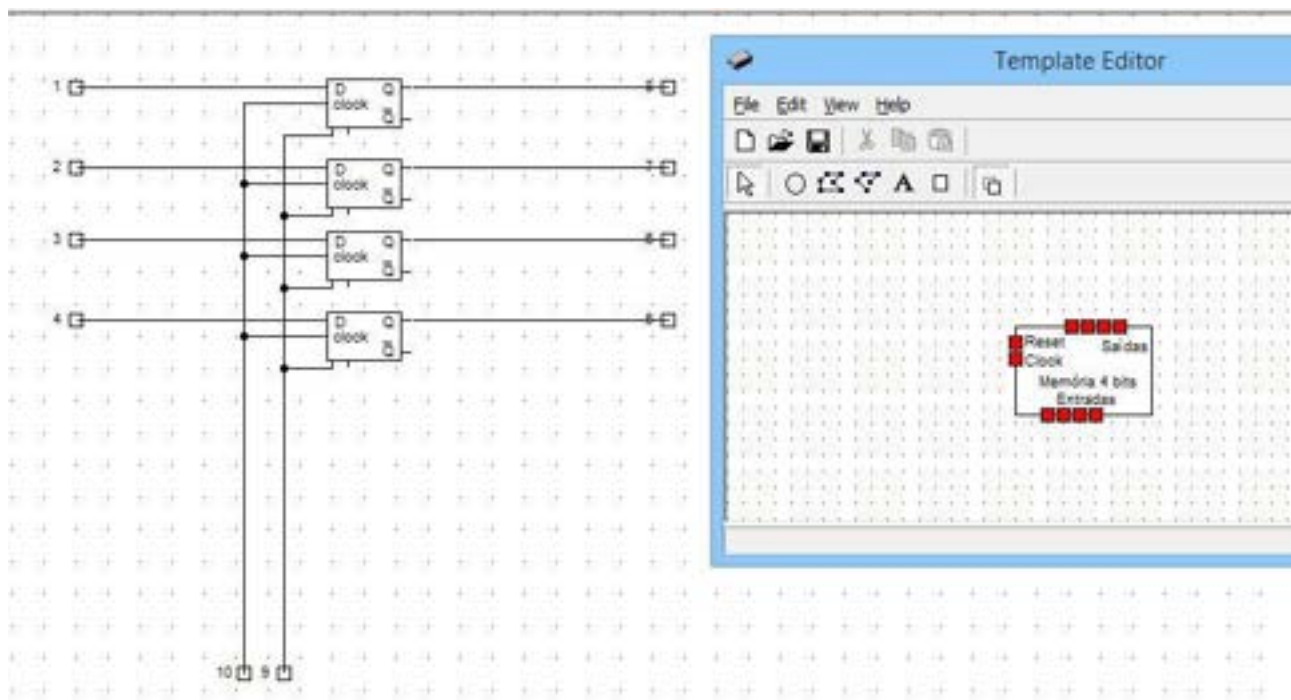




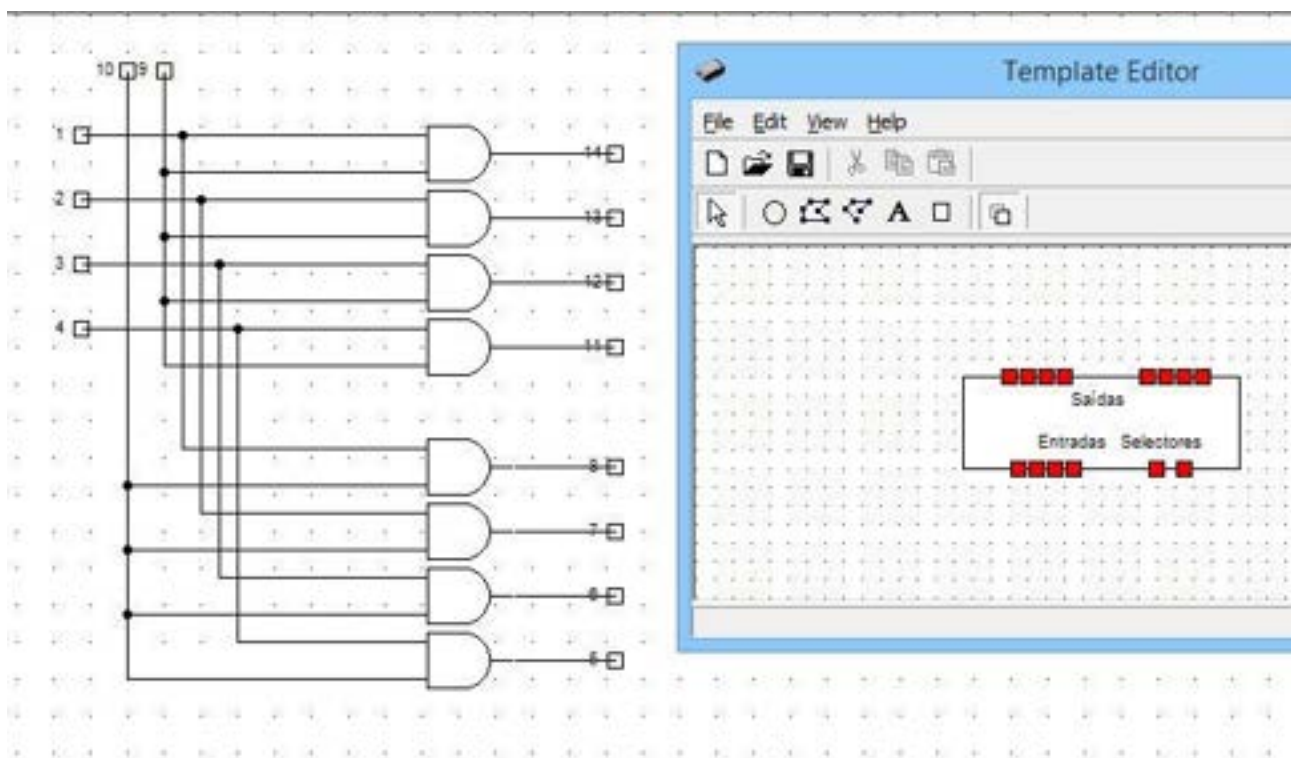
**Figura 38 – Alínea D – Componente memória 16 bits**



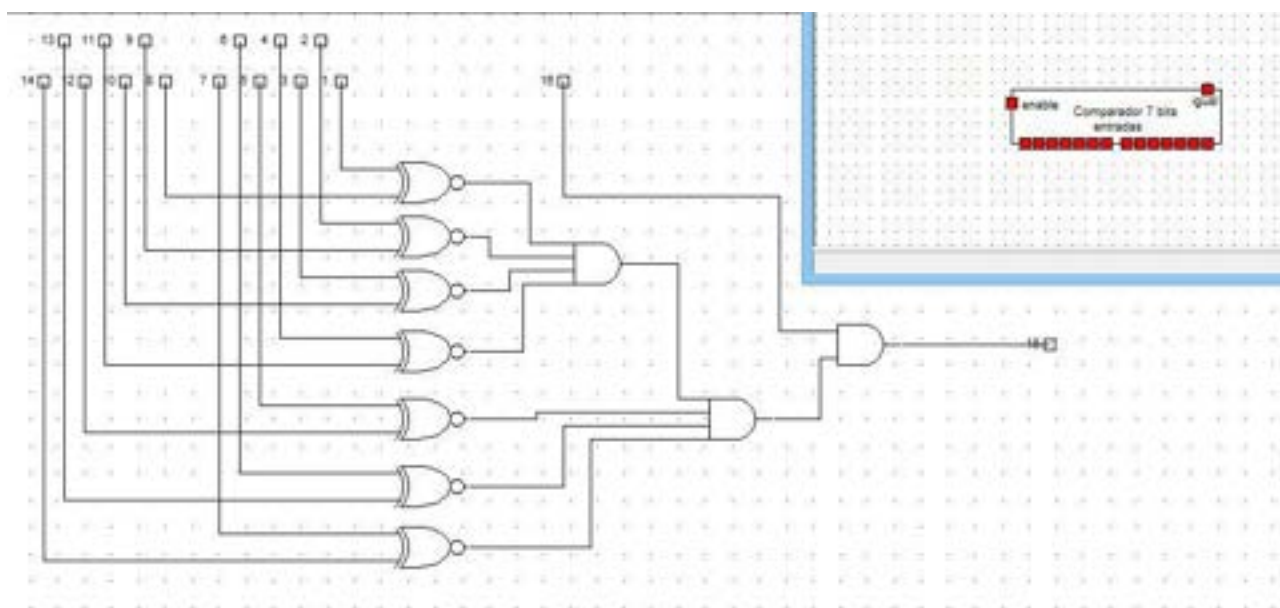
**Figura 39 – Alínea D – Componente conversor Endereço Memória para posição no array**



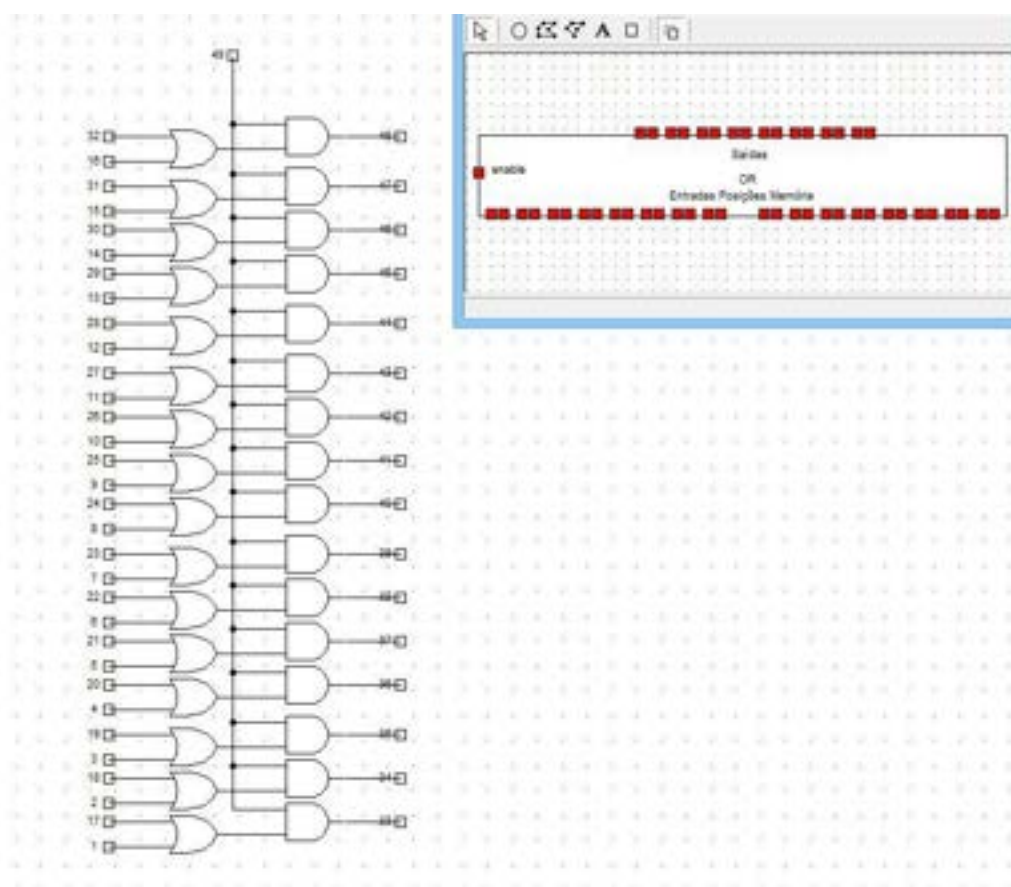
**Figura 40 – Alínea D – Componente Memória bits**



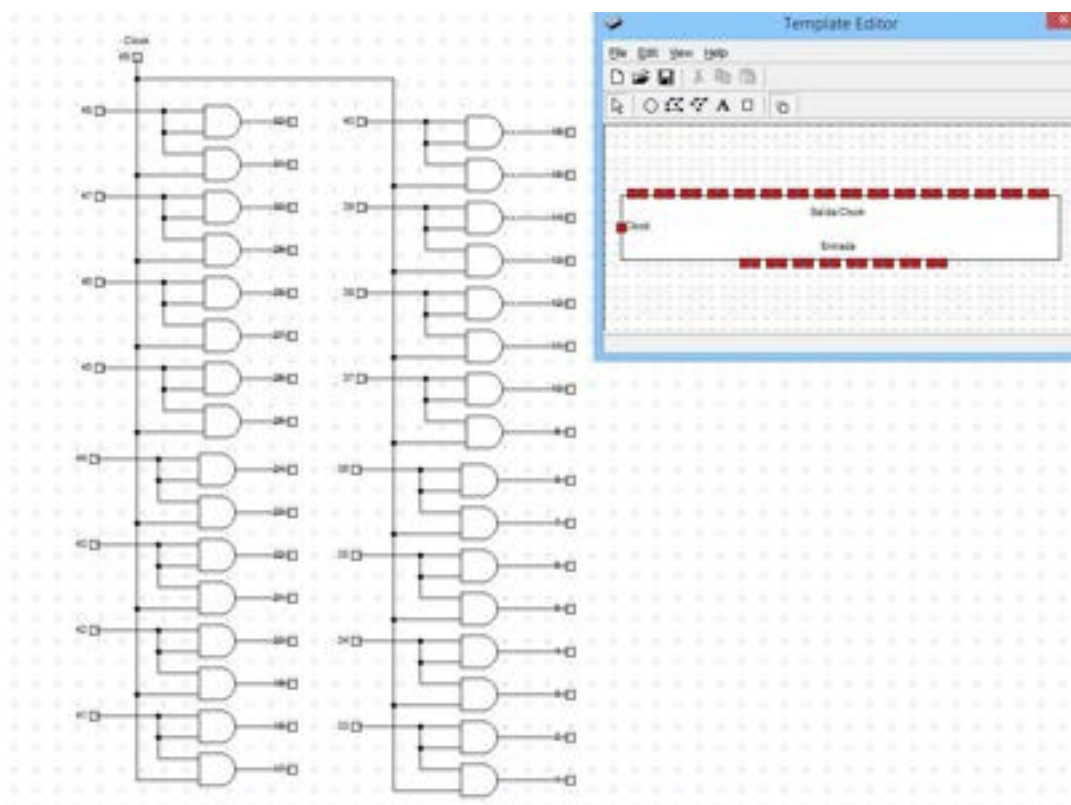
**Figura 41 – Alínea D – Componente Selector inicial**



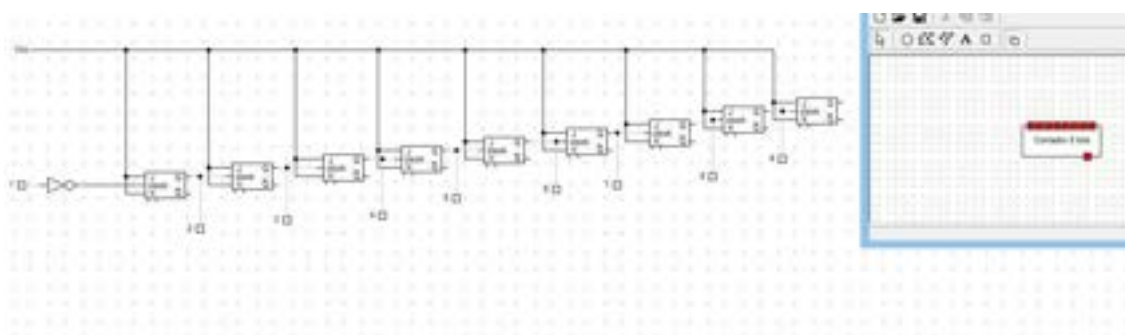
**Figura 42 – Alínea D – Componente Comparador de 7 bits**



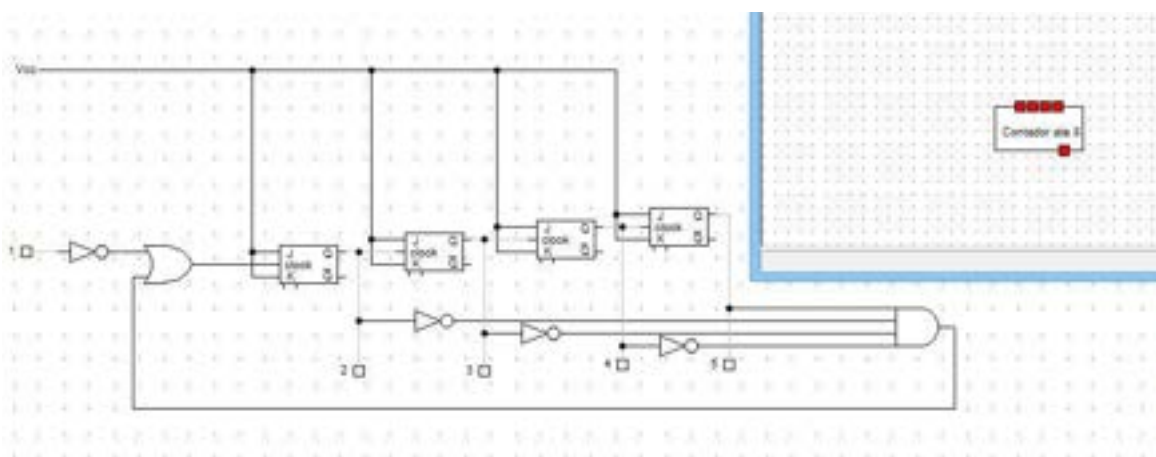
**Figura 43 – Alínea D – Componente Passagem OR Array**



**Figura 44 – Alínea D– Componente Escrita Array**

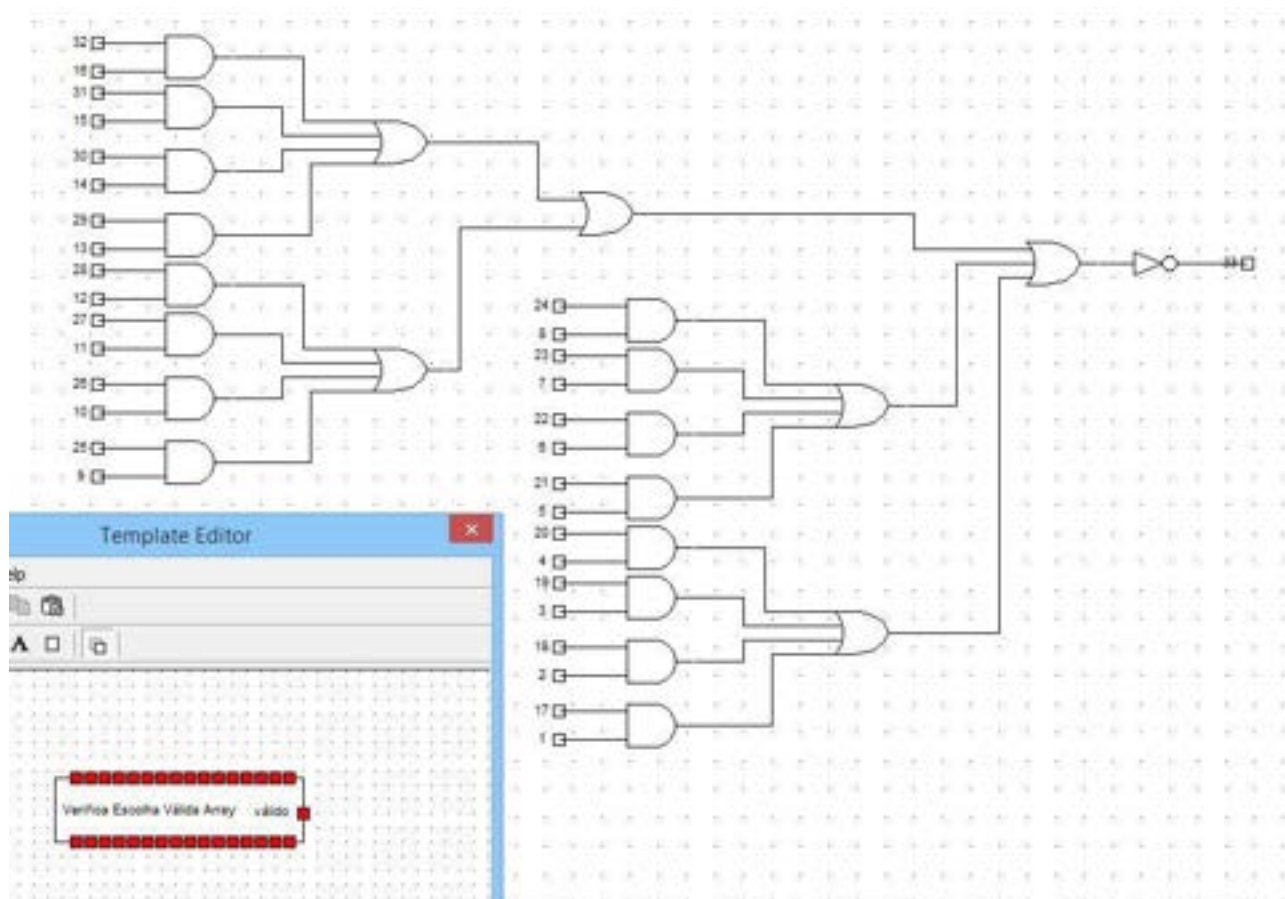


**Figura 45 – Alínea D– Componente Contador 8 bits (jogadas)**

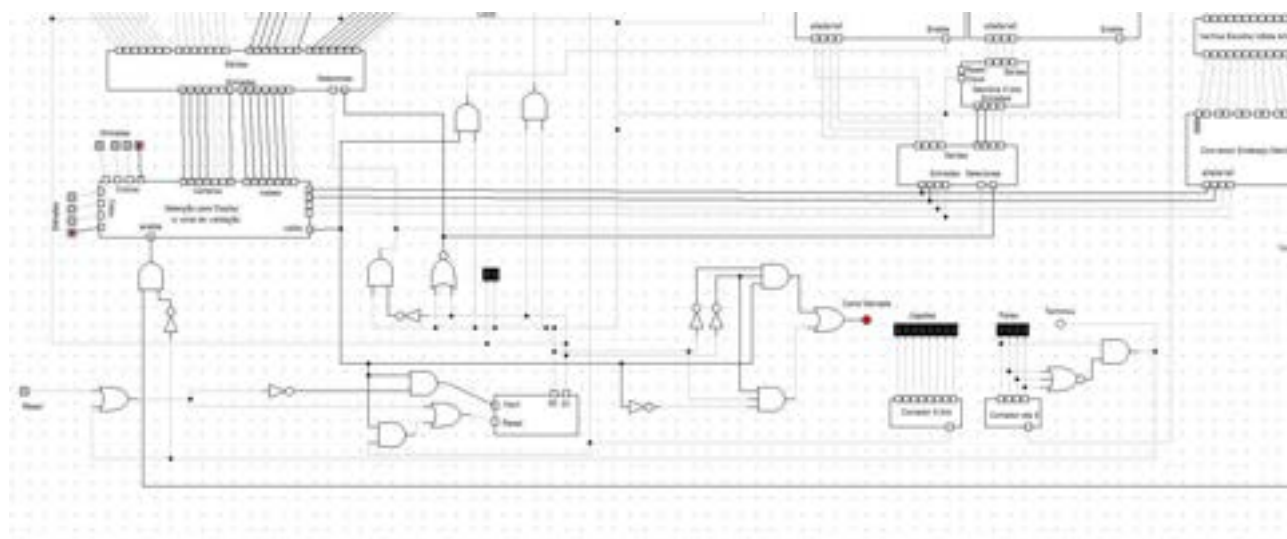


**Figura 46 – Alínea D– Componente Contador até 8 (pares)**

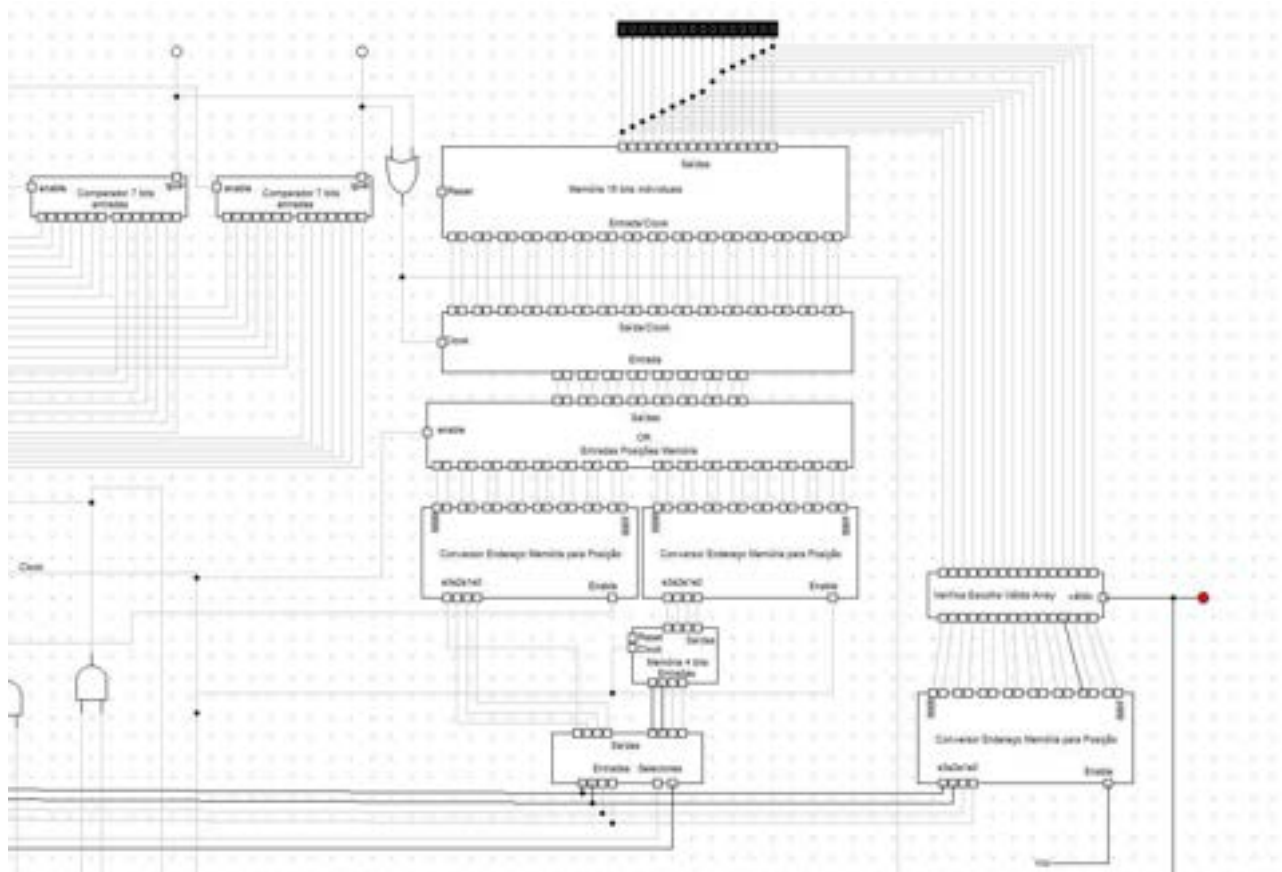




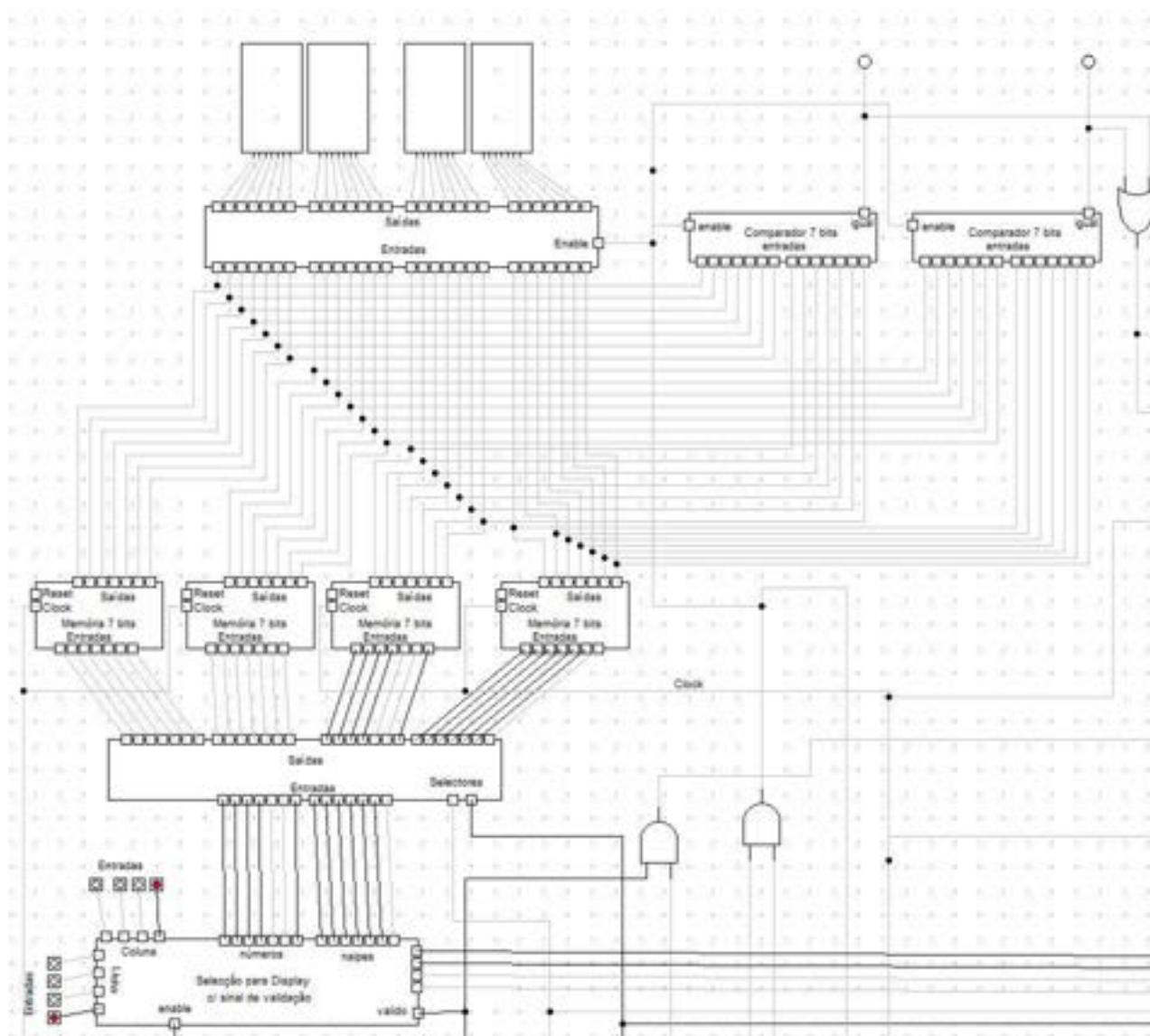
**Figura 47 – Alínea D – Componente valida entrada em relação ao Array**



**Figura 48 – Alínea D – Circuito Final (parcial)**



**Figura 49 – Alínea D – Circuito Final (parcial)**



**Figura 50 – Alínea D – Circuito Final (parcial)**

Teste de jogo completo

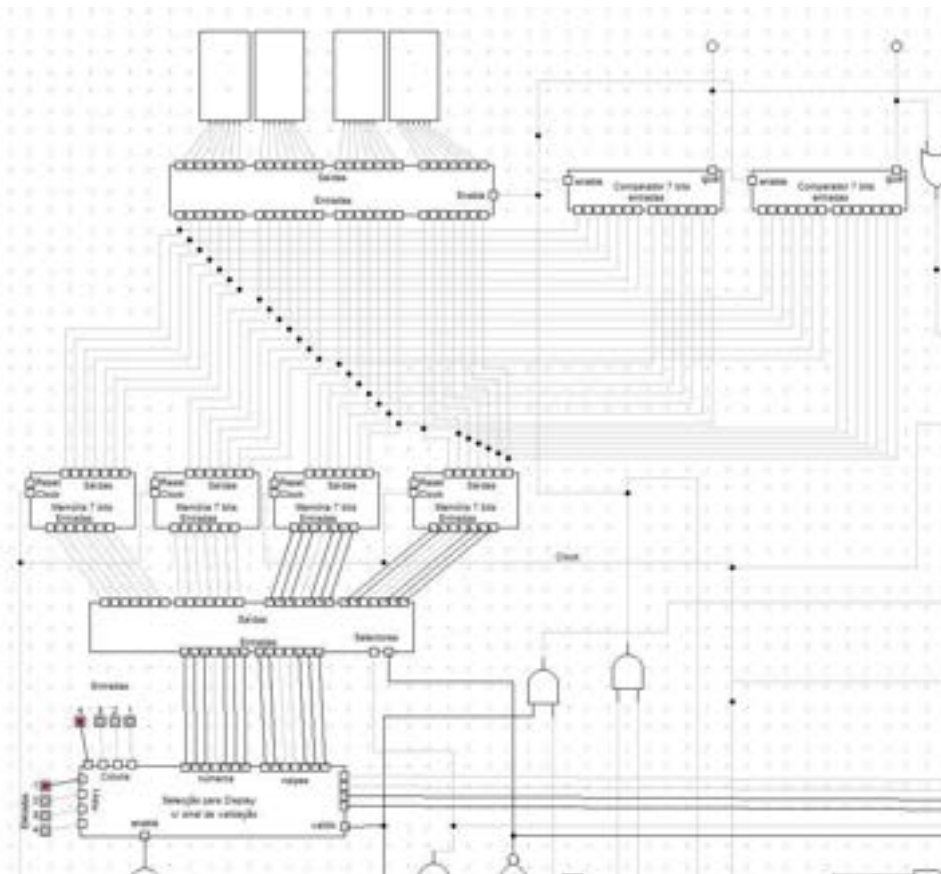


Figura 51 – Alínea D – Teste - Seleção posição 14 – Carta AP

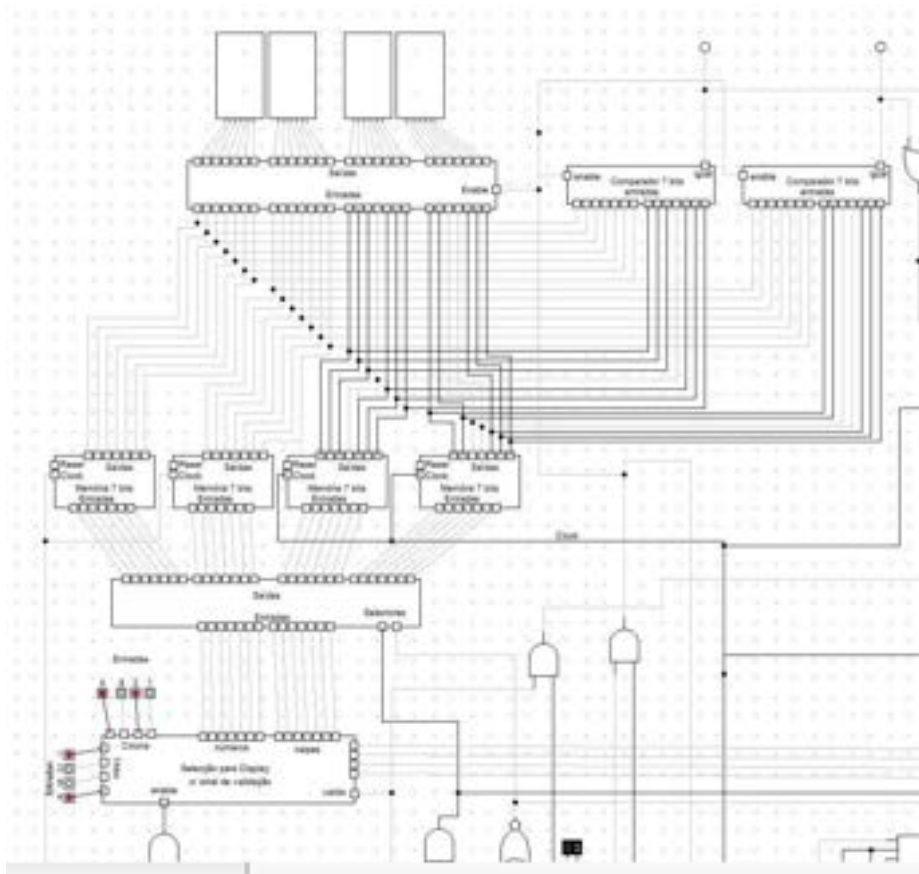
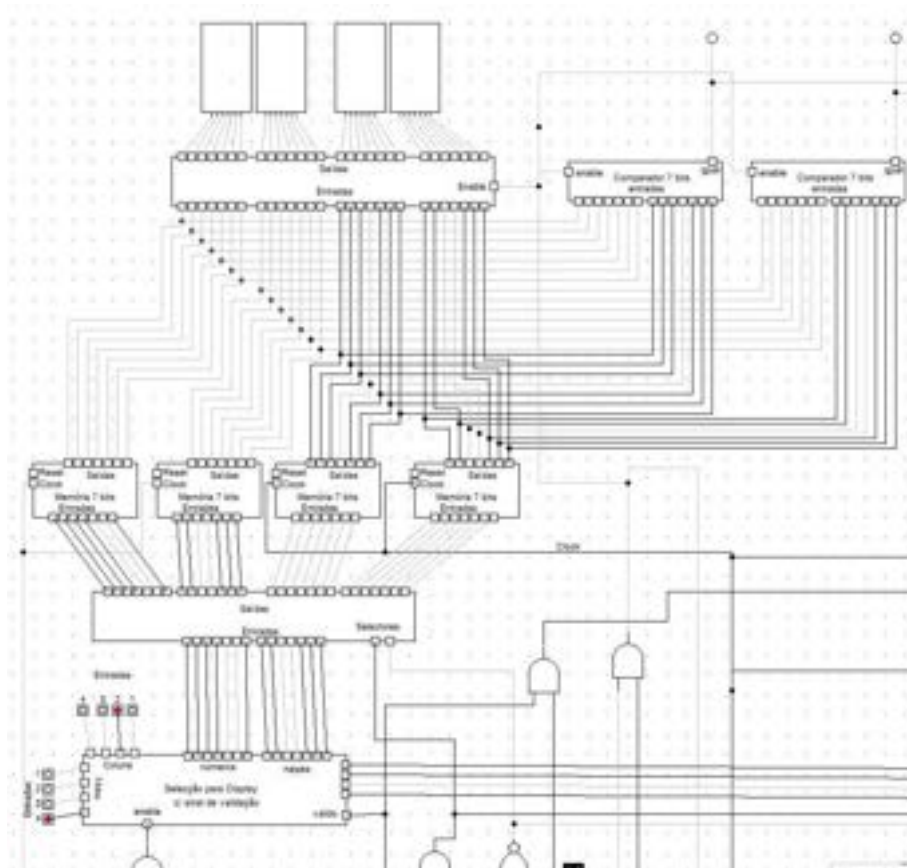
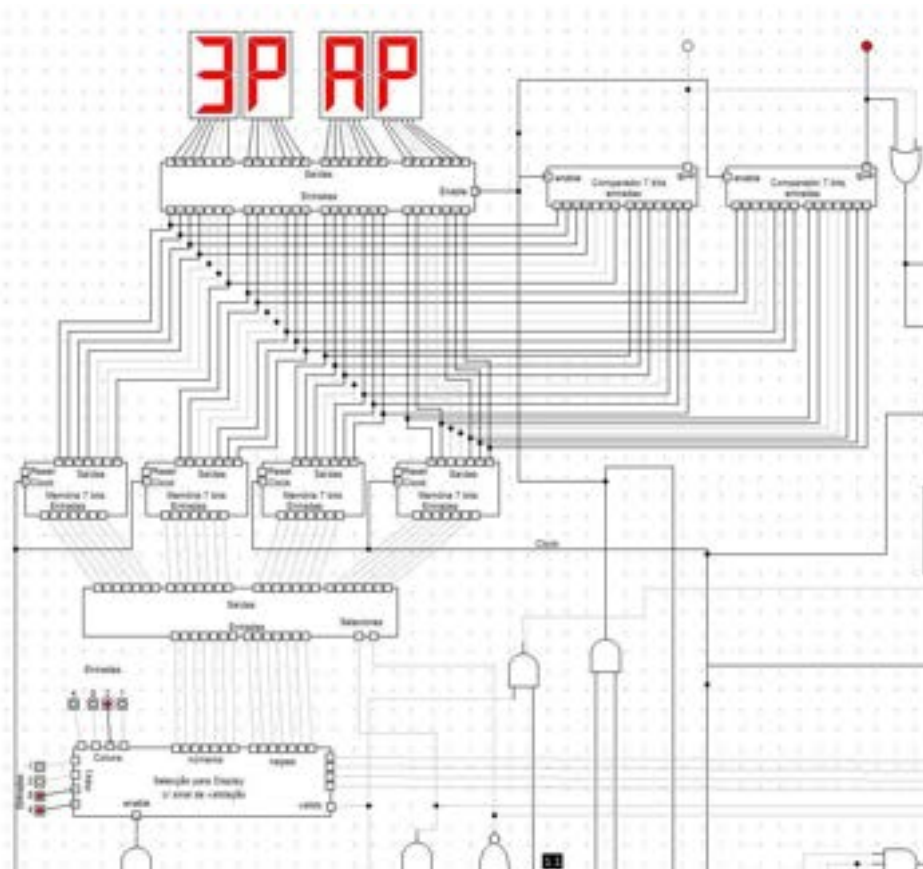


Figura 52 – Alínea D – Teste - Transição

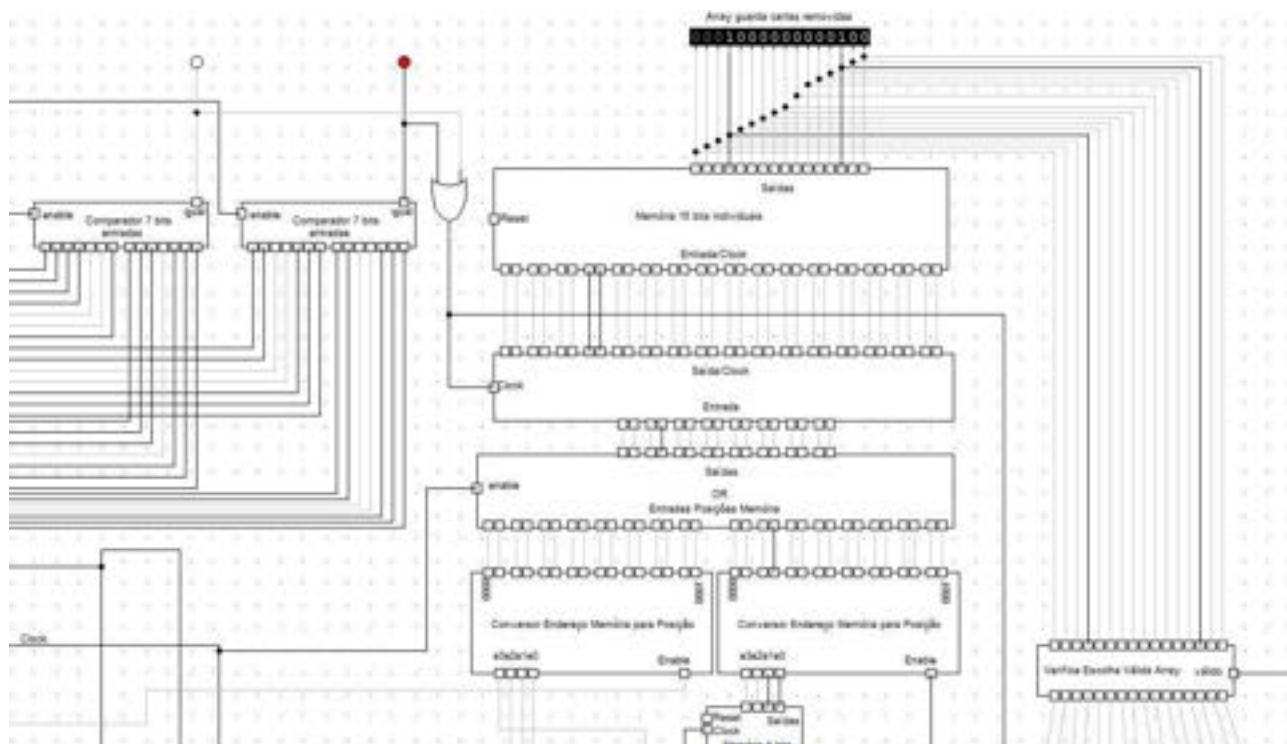


**Figura 53 – Alínea D – Teste - Selecção posição 42 – Carta 3P**

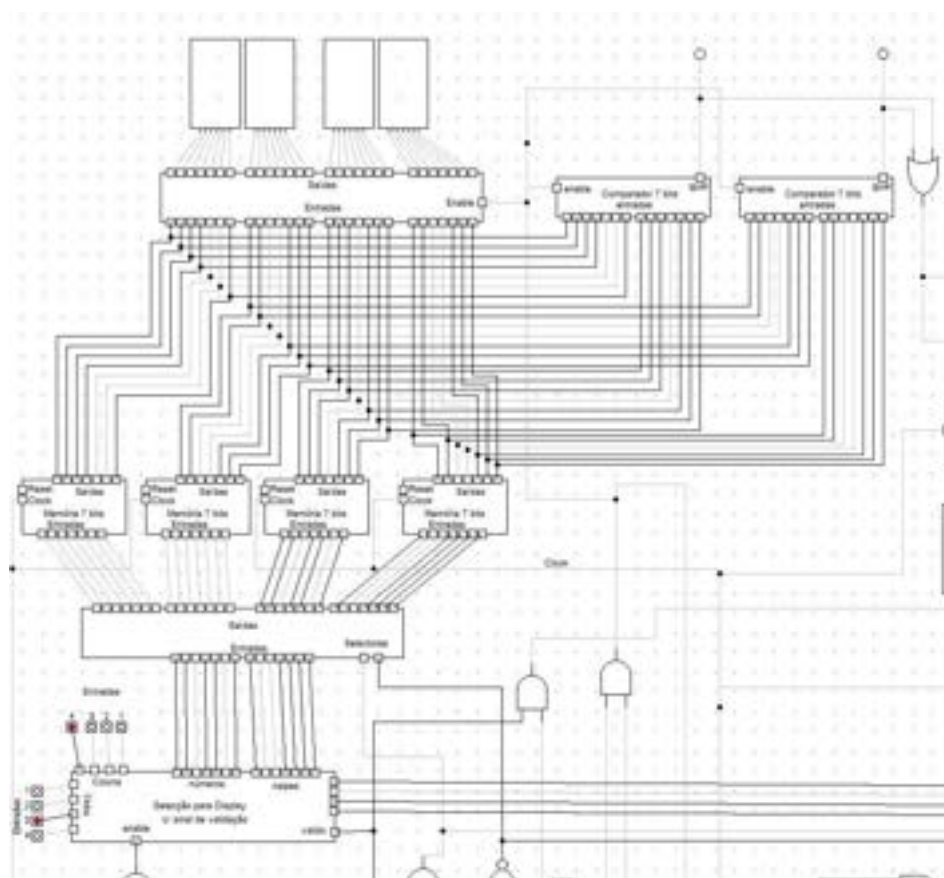


**Figura 54 – Alínea D – Teste – Display de Cartas**

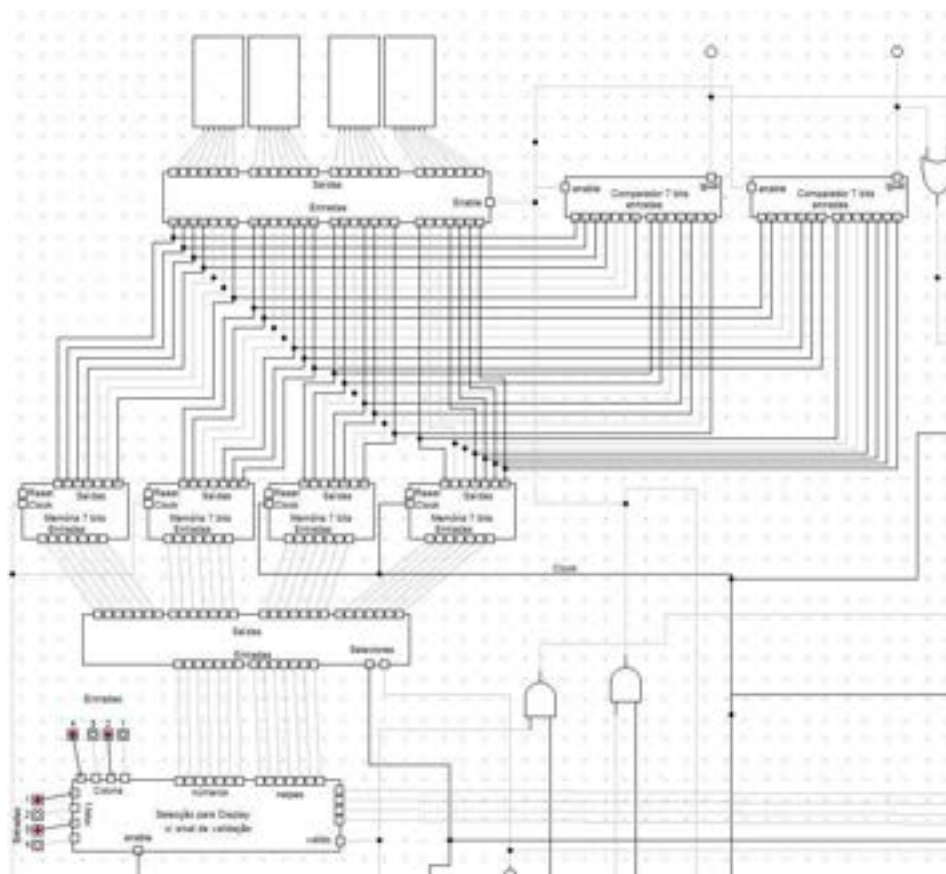




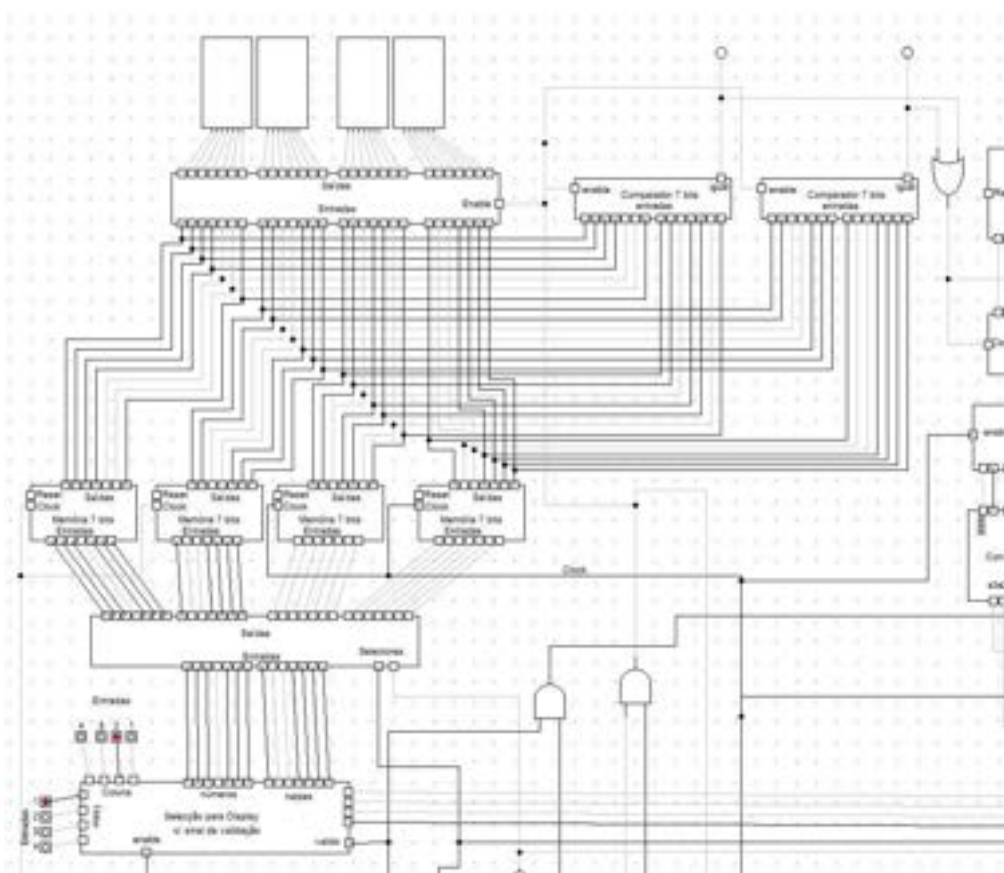
**Figura 55 – Alínea D – Teste – Guardar no Array**



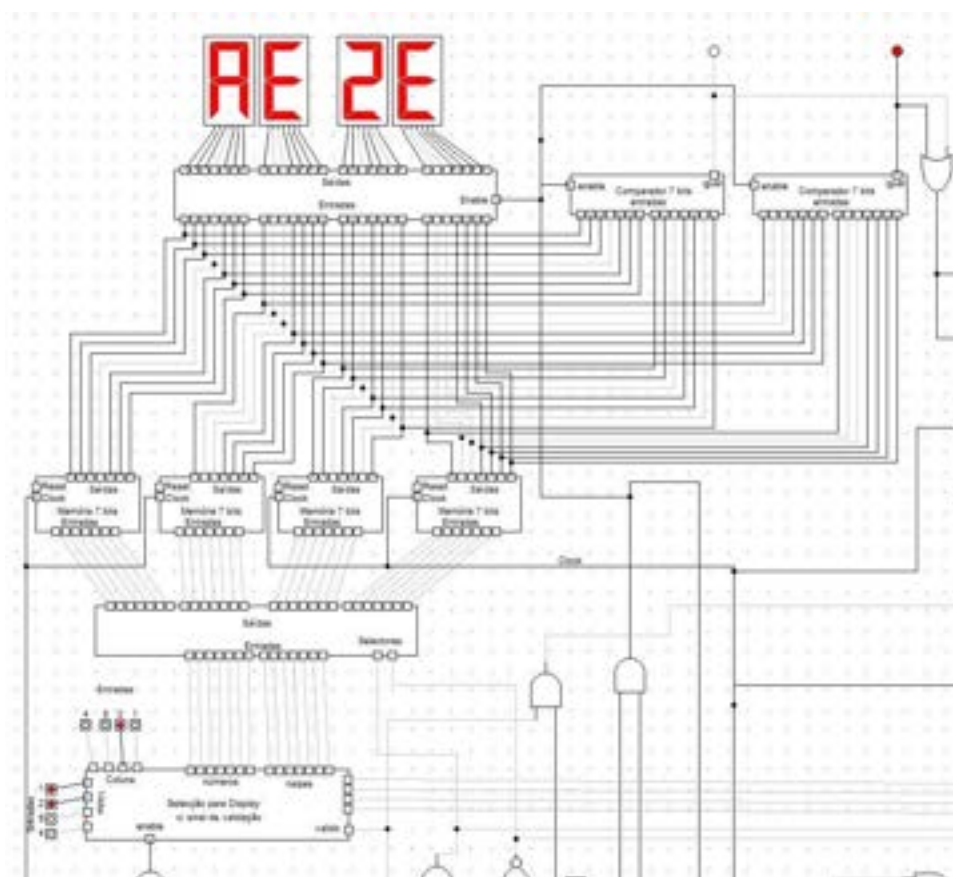
**Figura 56 – Alínea D – Teste - Seleção posição 34 – Carta 2E**



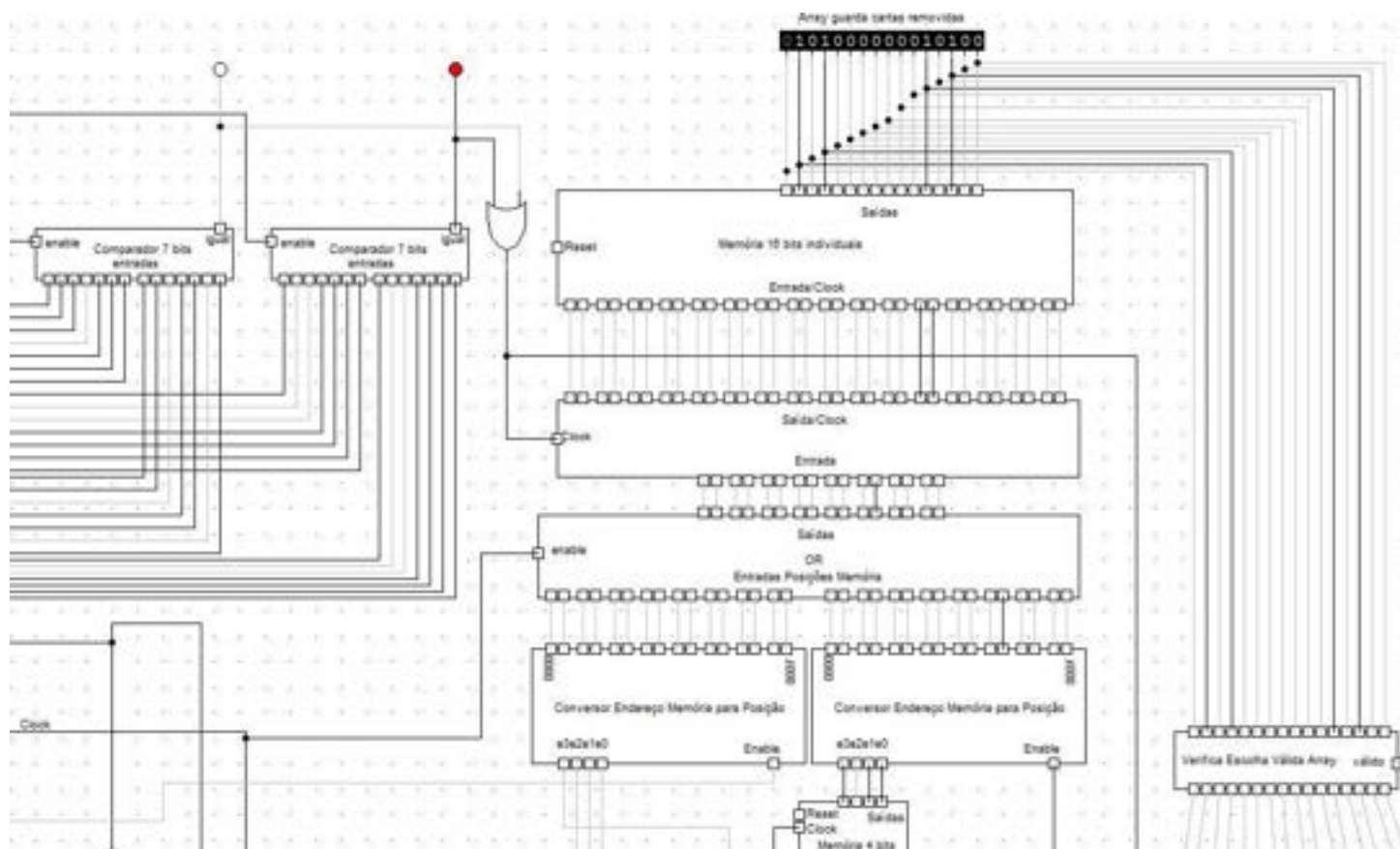
**Figura 57 – Alínea D – Transição**



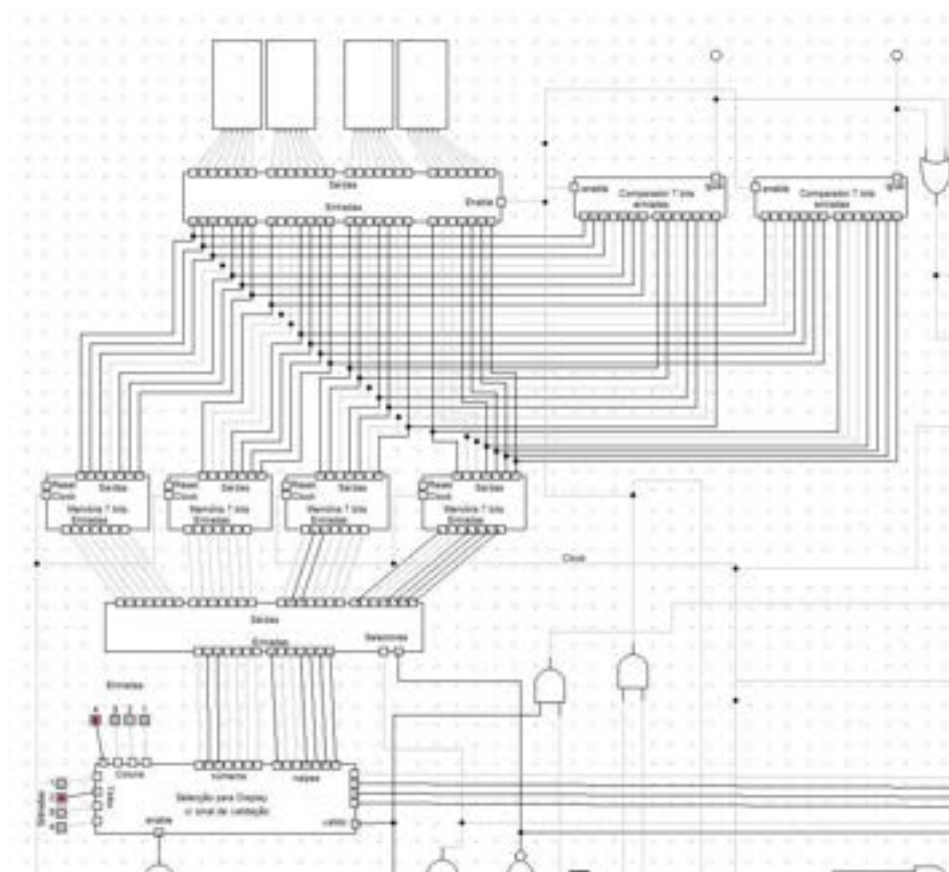
**Figura 58 – Alínea D – Teste - Seleção posição 12 – Carta AE**



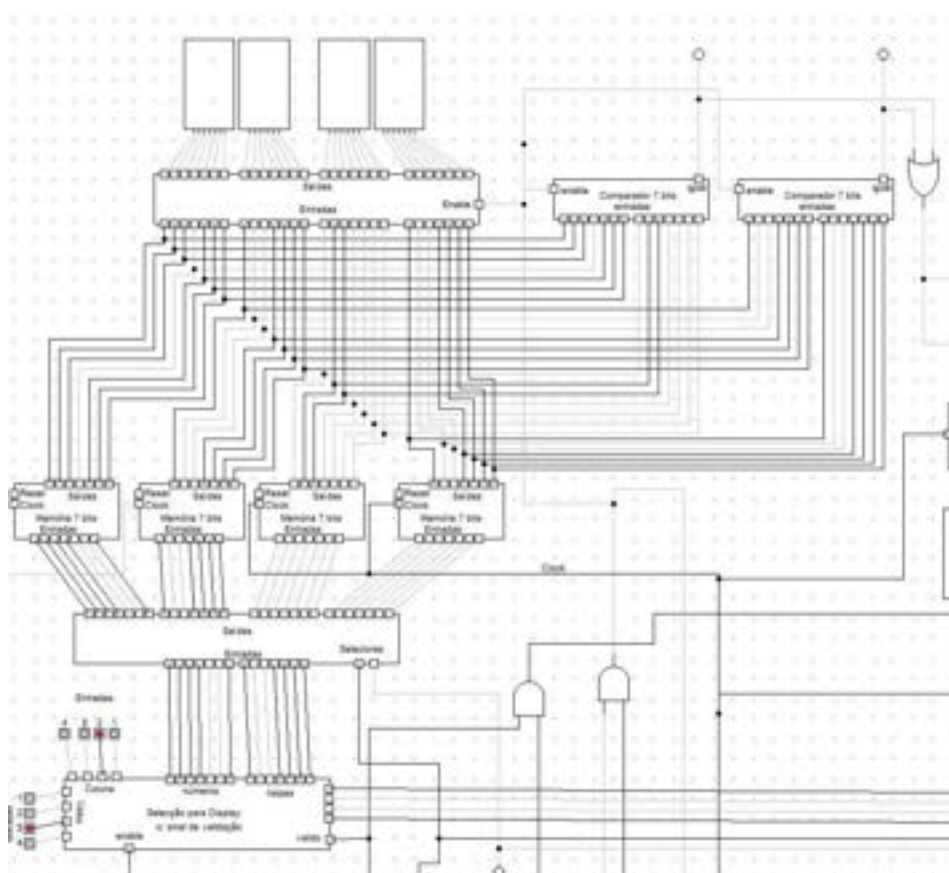
**Figura 59 – Alínea D – Teste – Display de Cartas**



**Figura 60 – Alínea D – Teste – Guardar no Array**

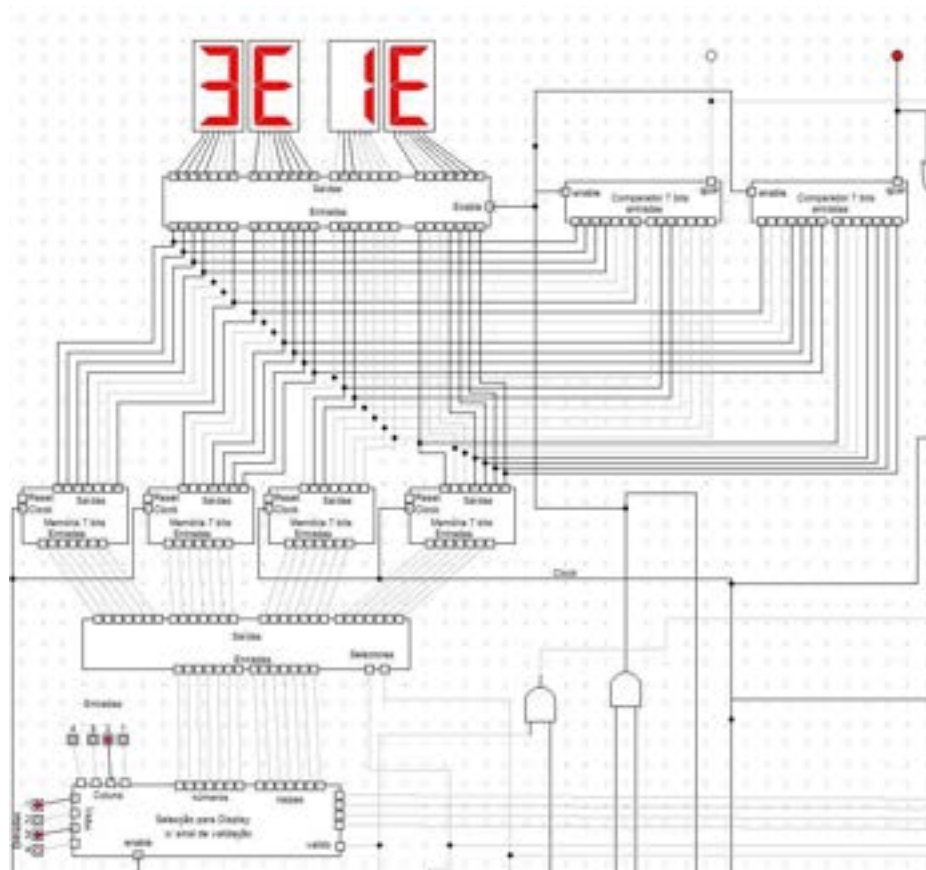


**Figura 61 – Alínea D – Teste - Seleção posição 24 – Carta 1E**

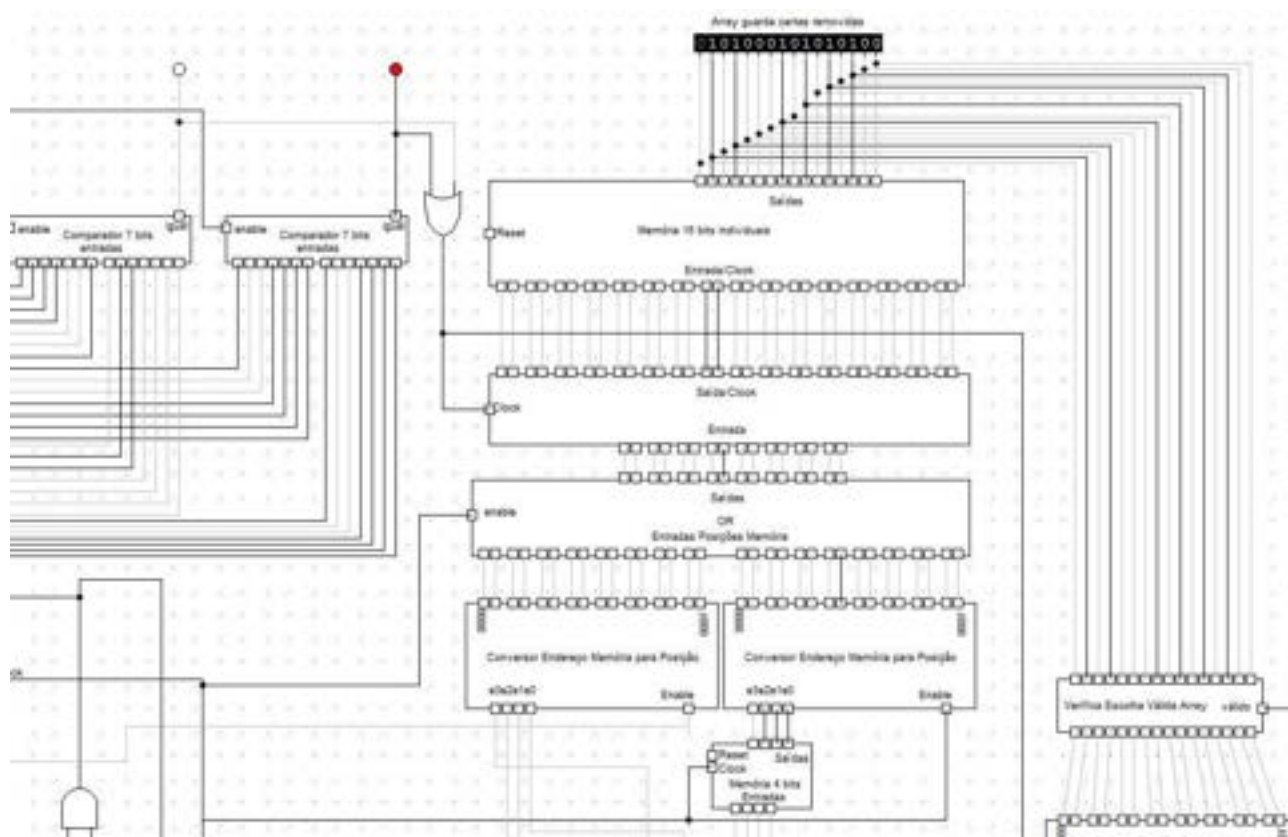


**Figura 62 – Alínea D – Teste - Seleção posição 32 – Carta 3E**



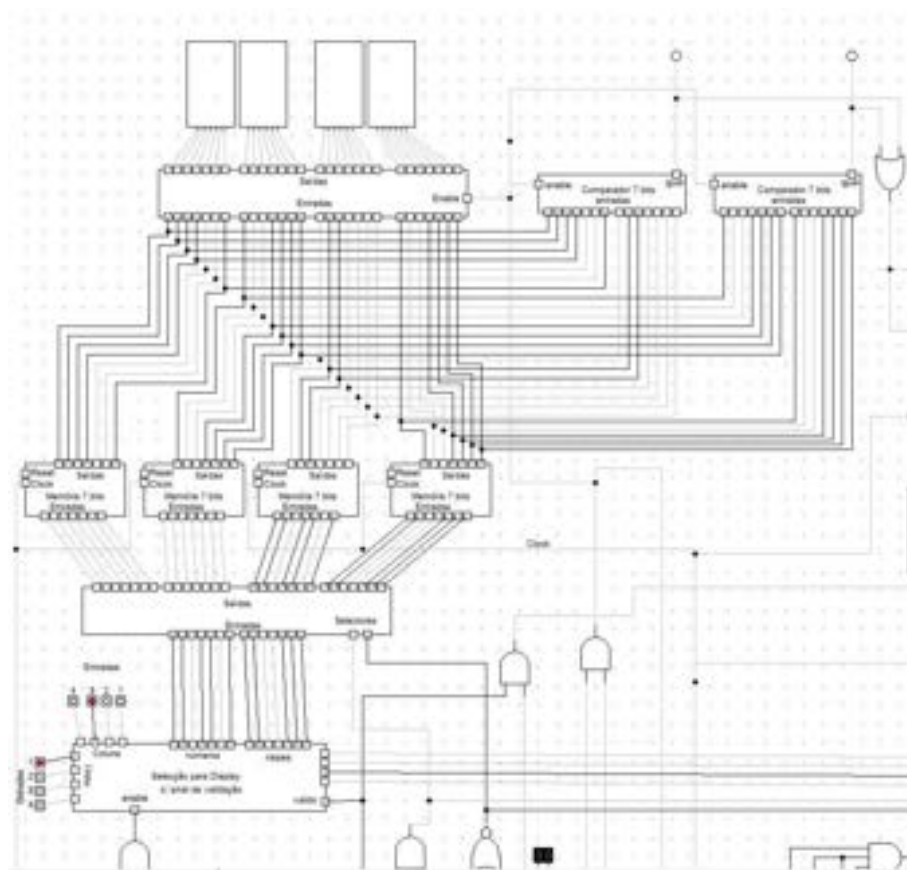


**Figura 63 – Alínea D – Teste – Display de Cartas**

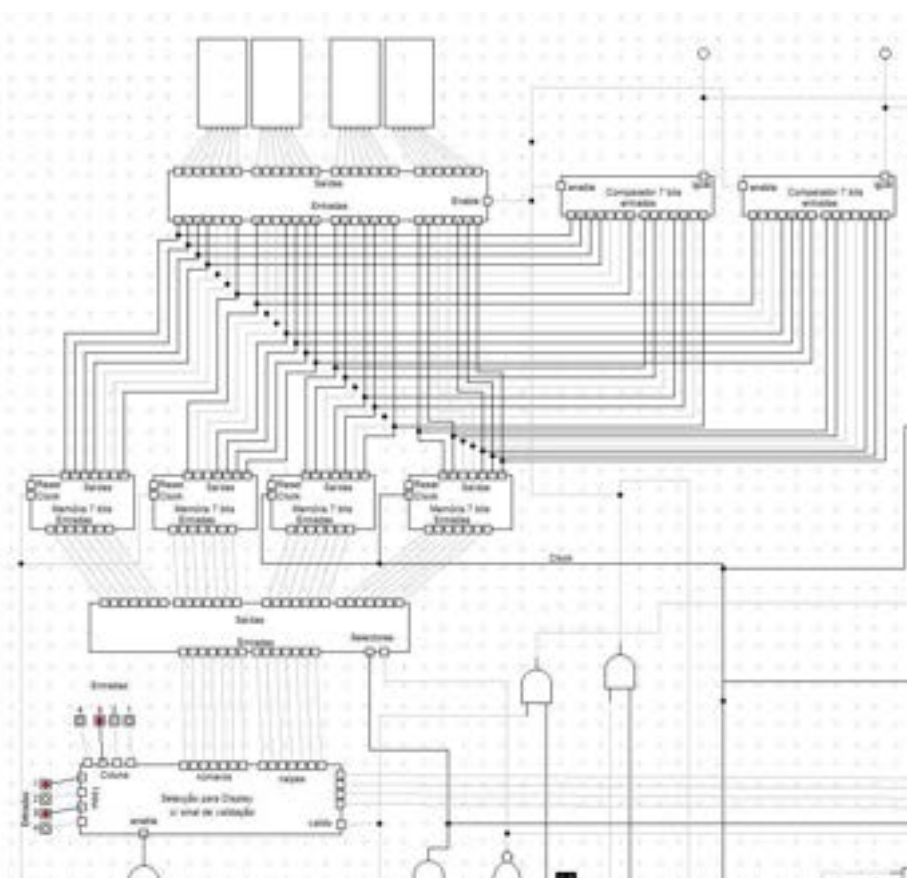


**Figura 64 – Alínea D – Teste – Guardar no Array**

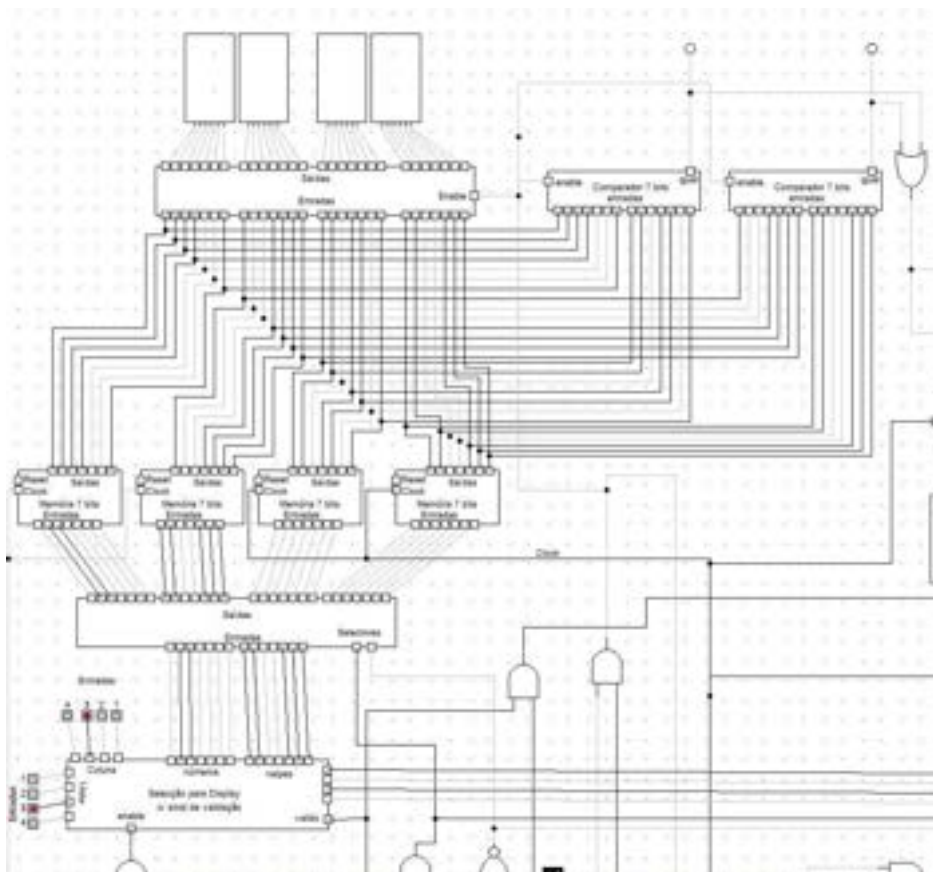




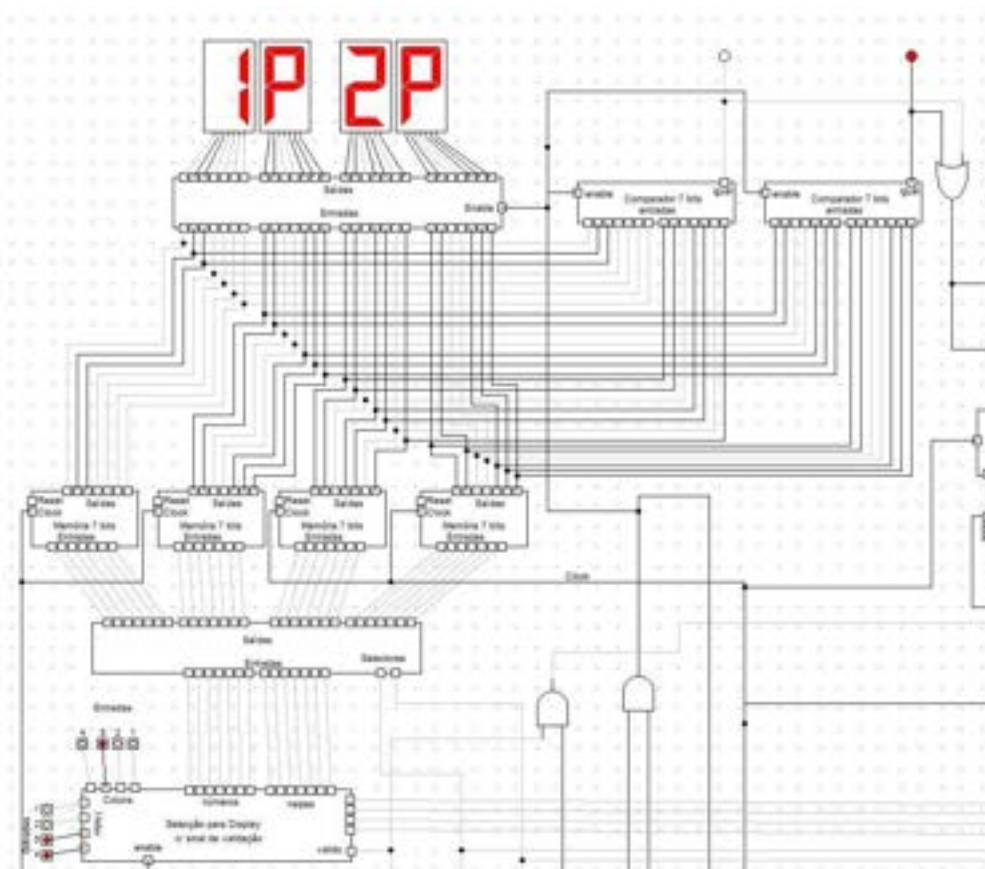
**Figura 65 – Alínea D – Teste - Seleção posição 13 – Carta 2P**



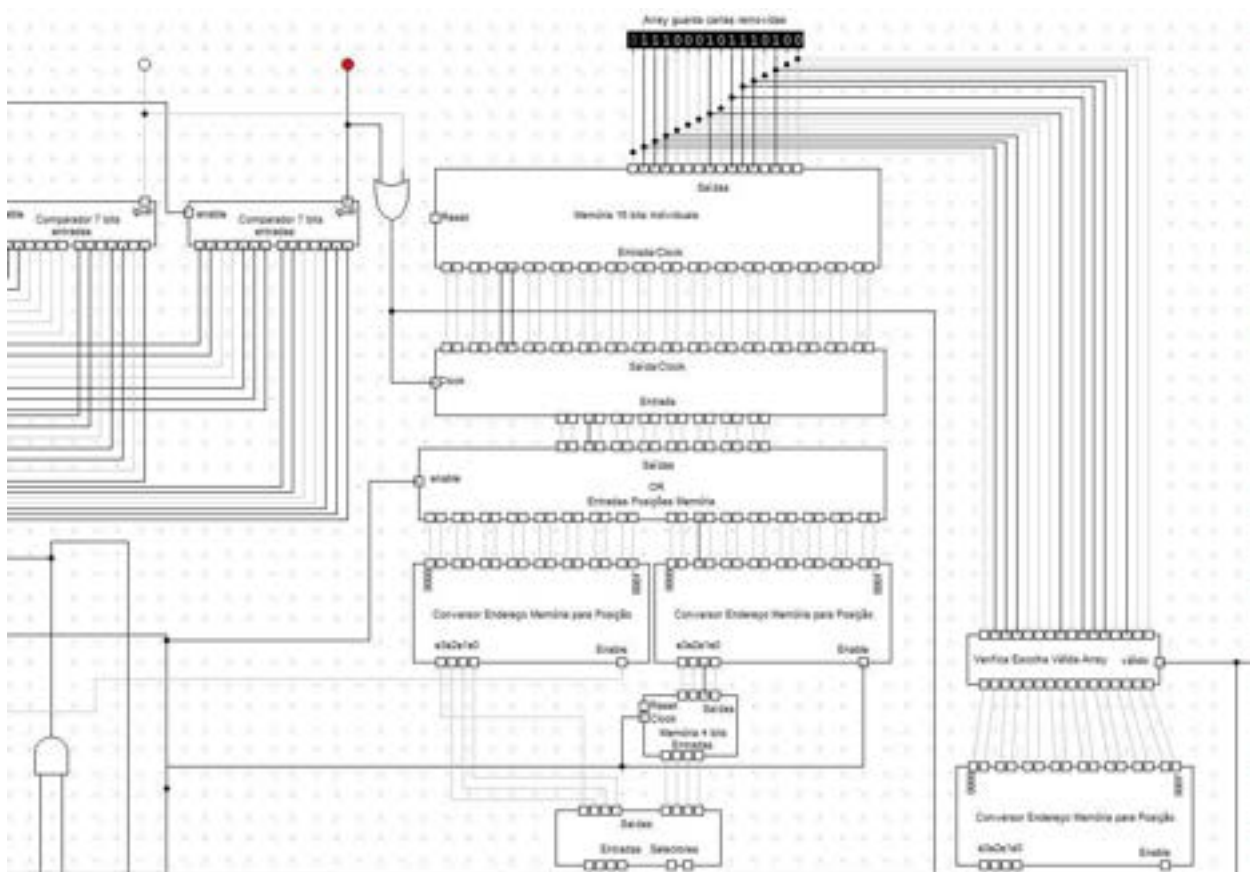
**Figura 66 – Alínea D – Transição**



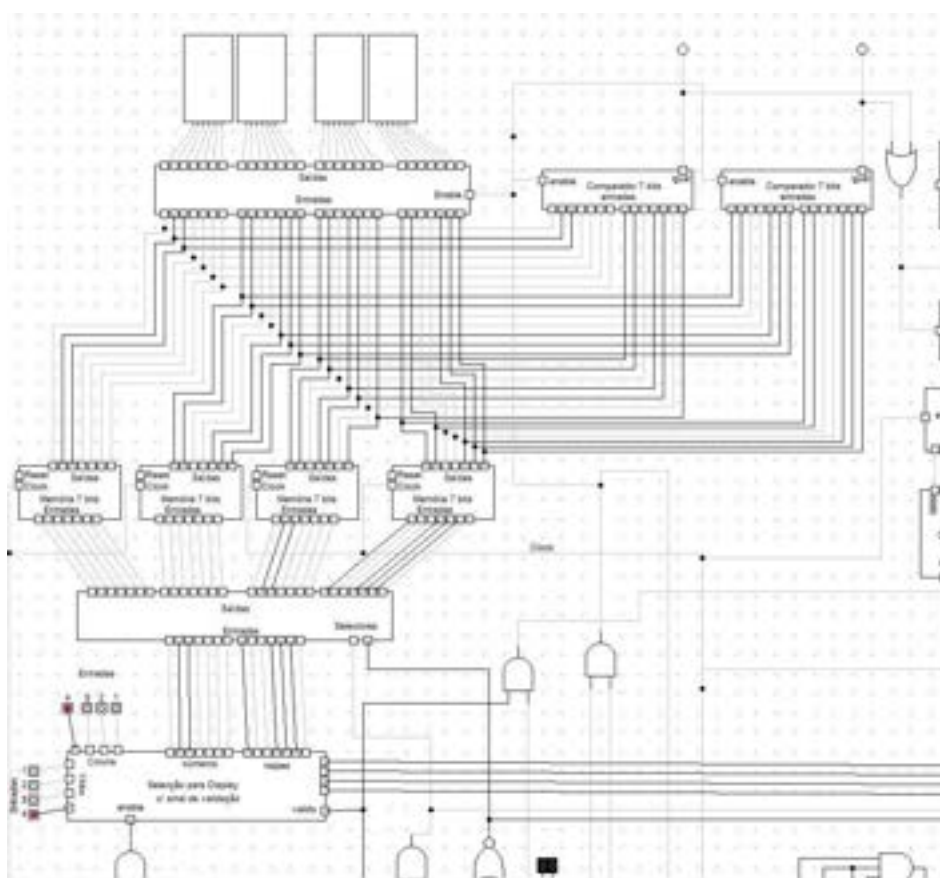
**Figura 67 – Alínea D – Teste - Seleção posição 33 – Carta 1P**



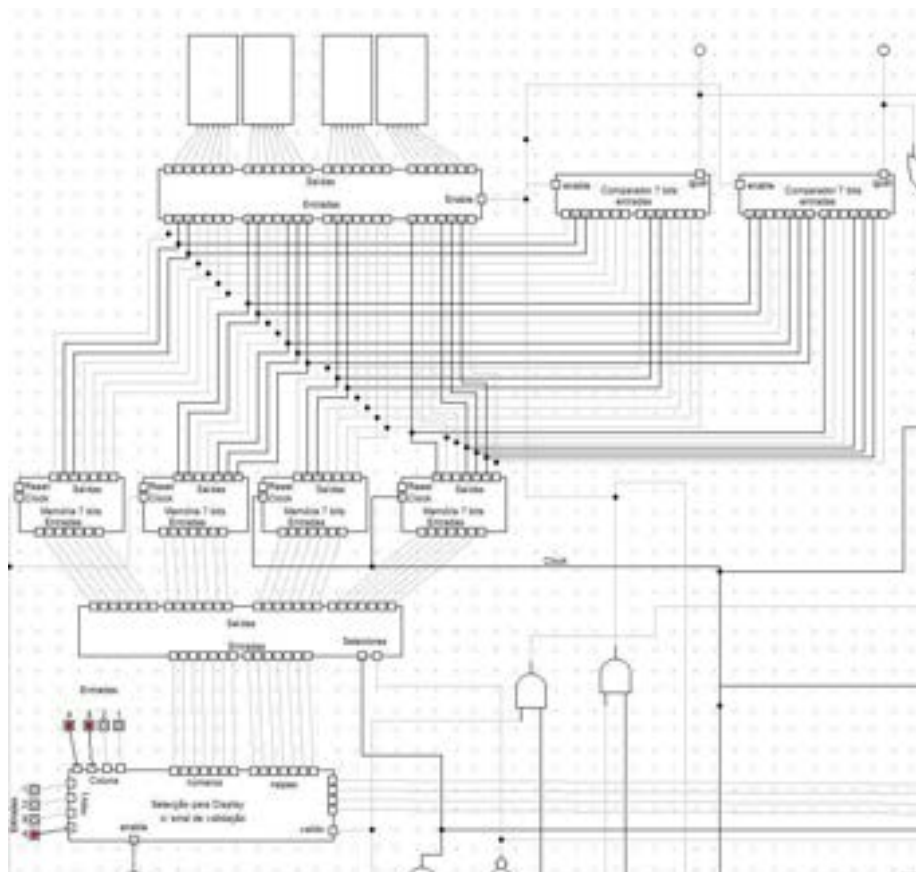
**Figura 68 – Alínea D – Teste – Display de Cartas**



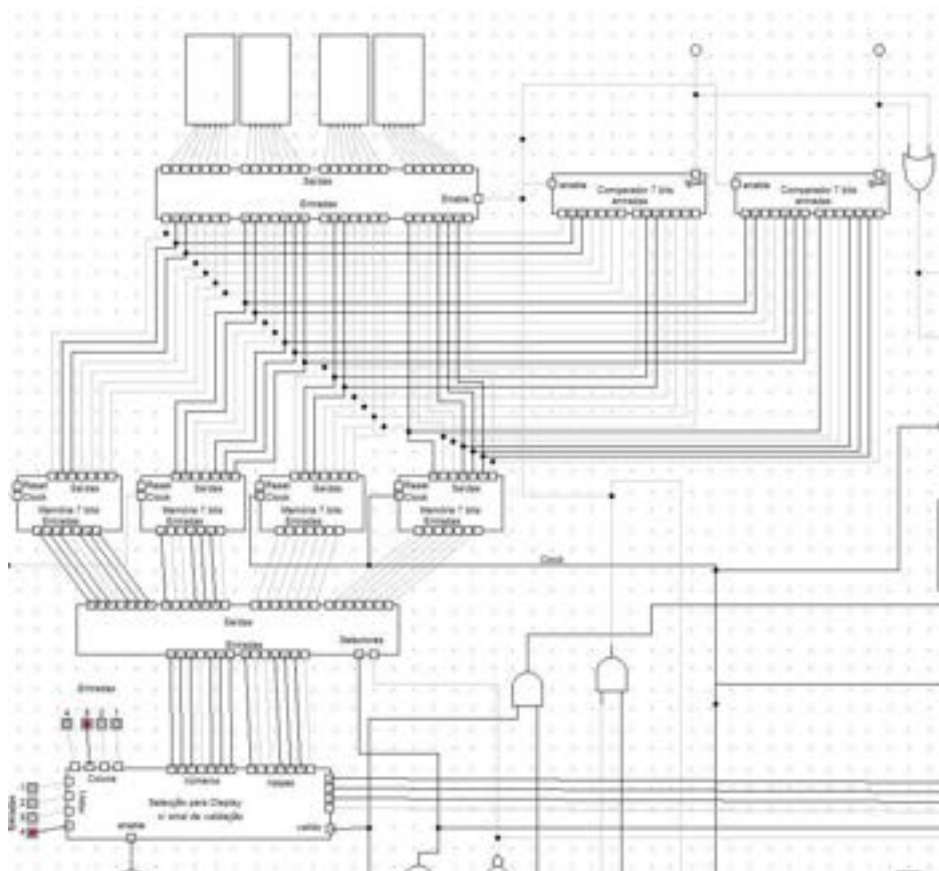
**Figura 69 – Alínea D – Teste – Guardar no Array**



**Figura 70 – Alínea D – Teste - Seleção posição 44 – Carta 1C**

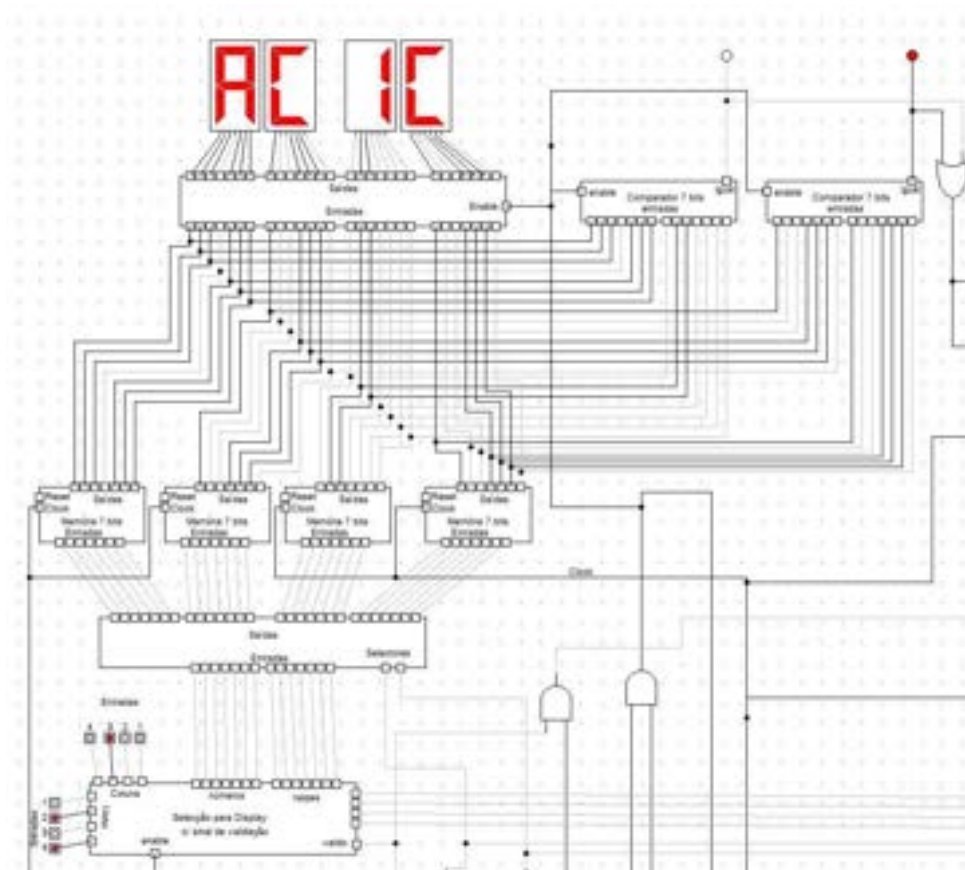


**Figura 71 – Alínea D – Transição**

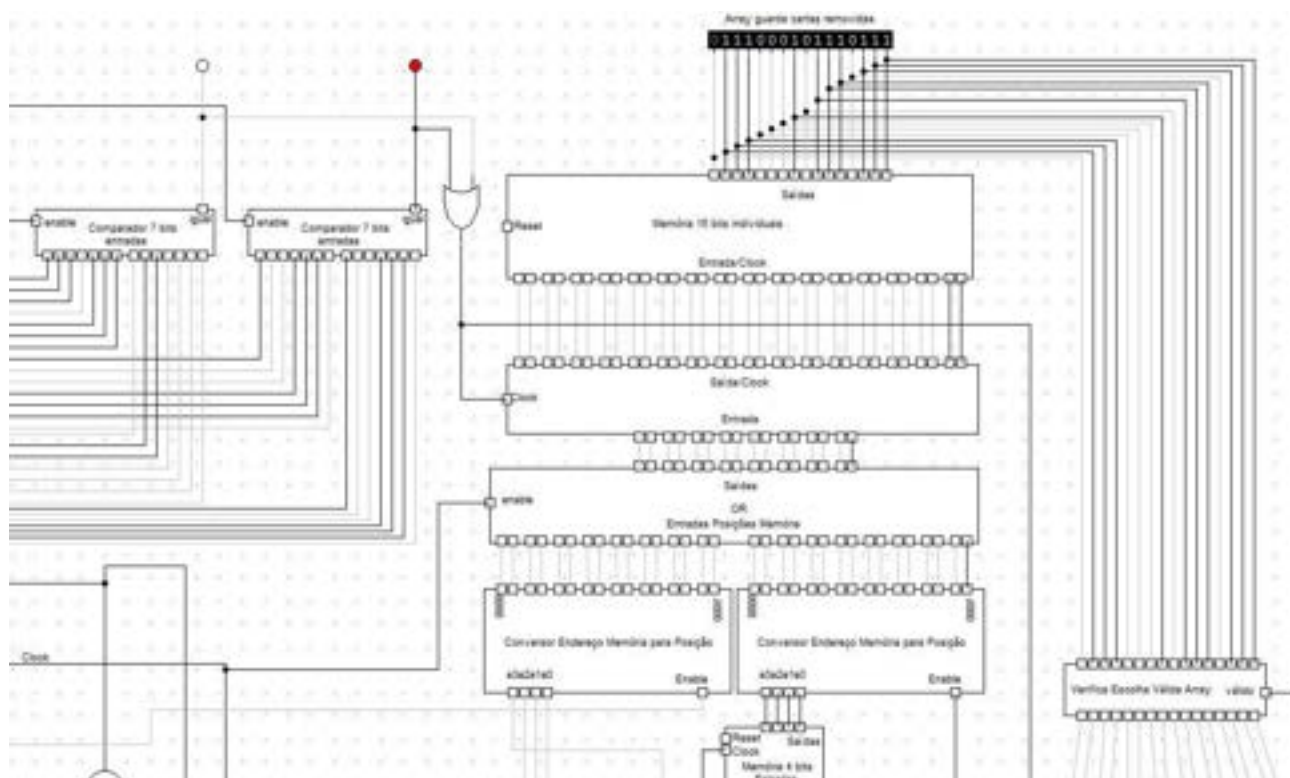


**Figura 72 – Alínea D – Teste - Seleção posição 43 – Carta AC**



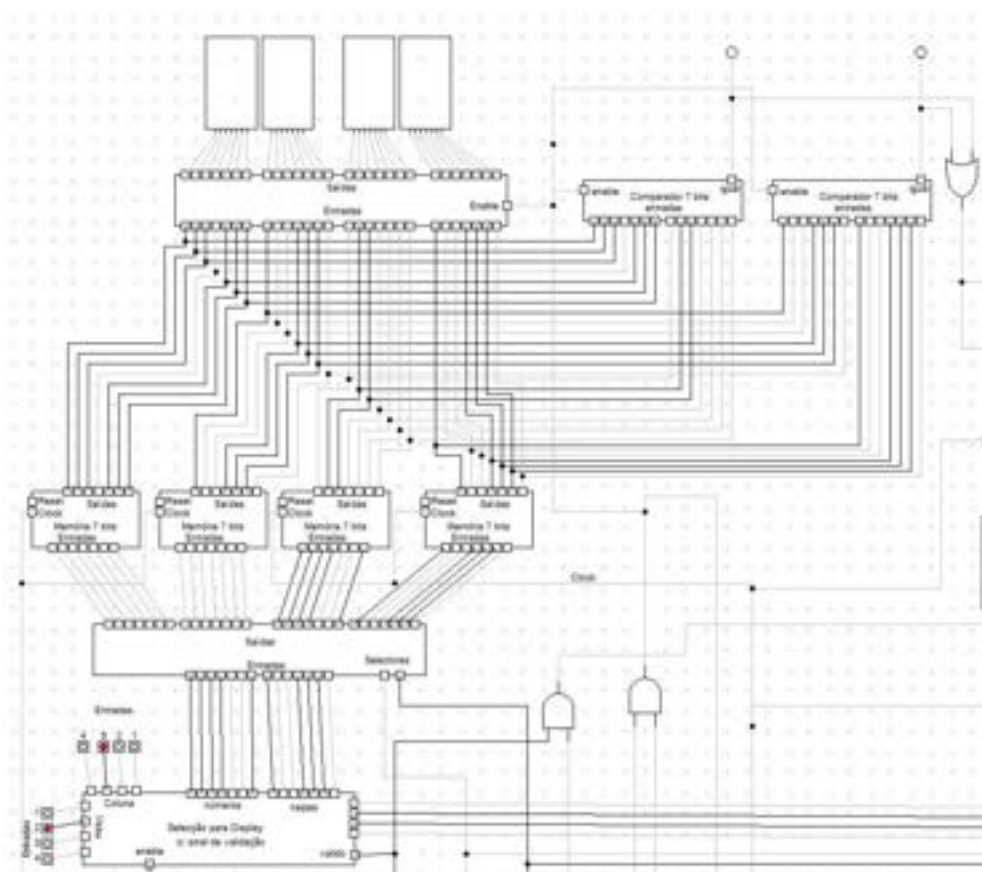


**Figura 73 – Alínea D – Teste – Display de Cartas**

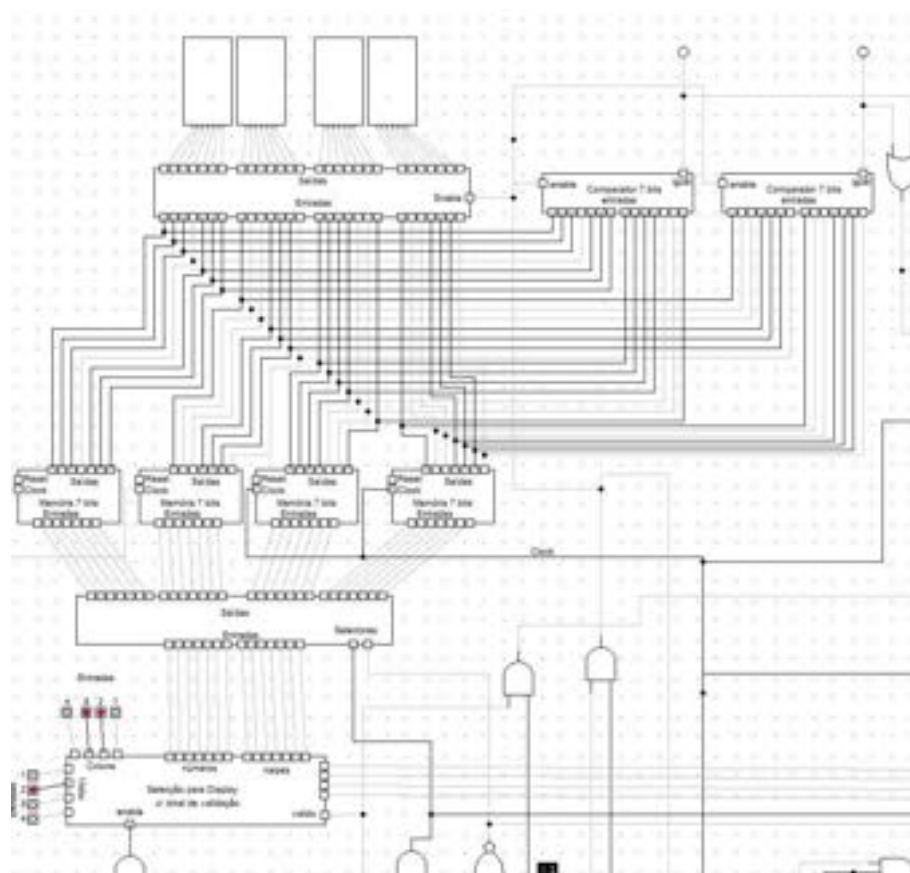


**Figura 74 – Alínea D – Teste – Guardar no Array**

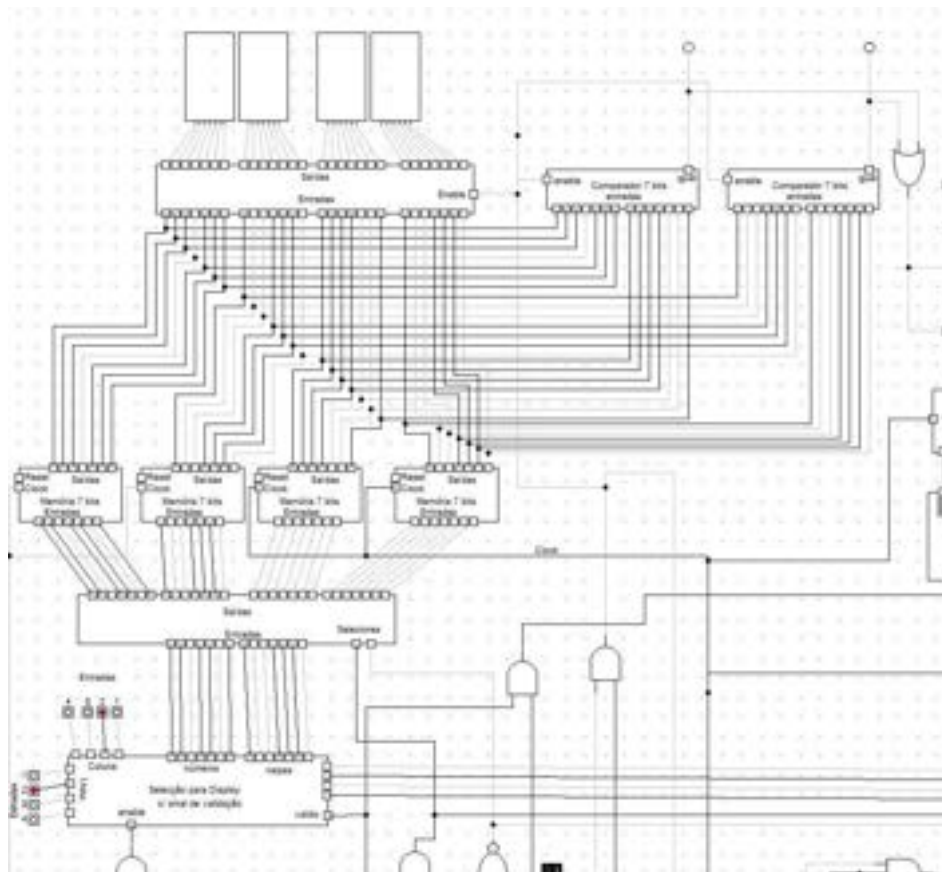




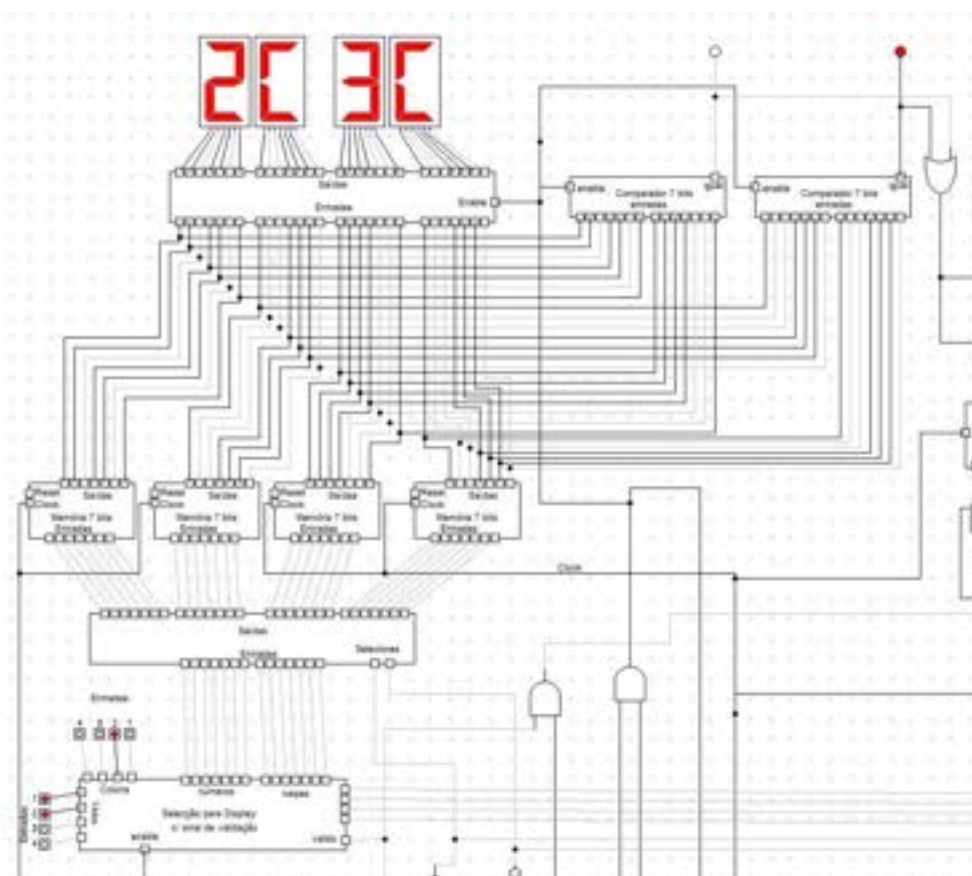
**Figura 75 – Alínea D – Teste - Seleção posição 23 – Carta 3C**



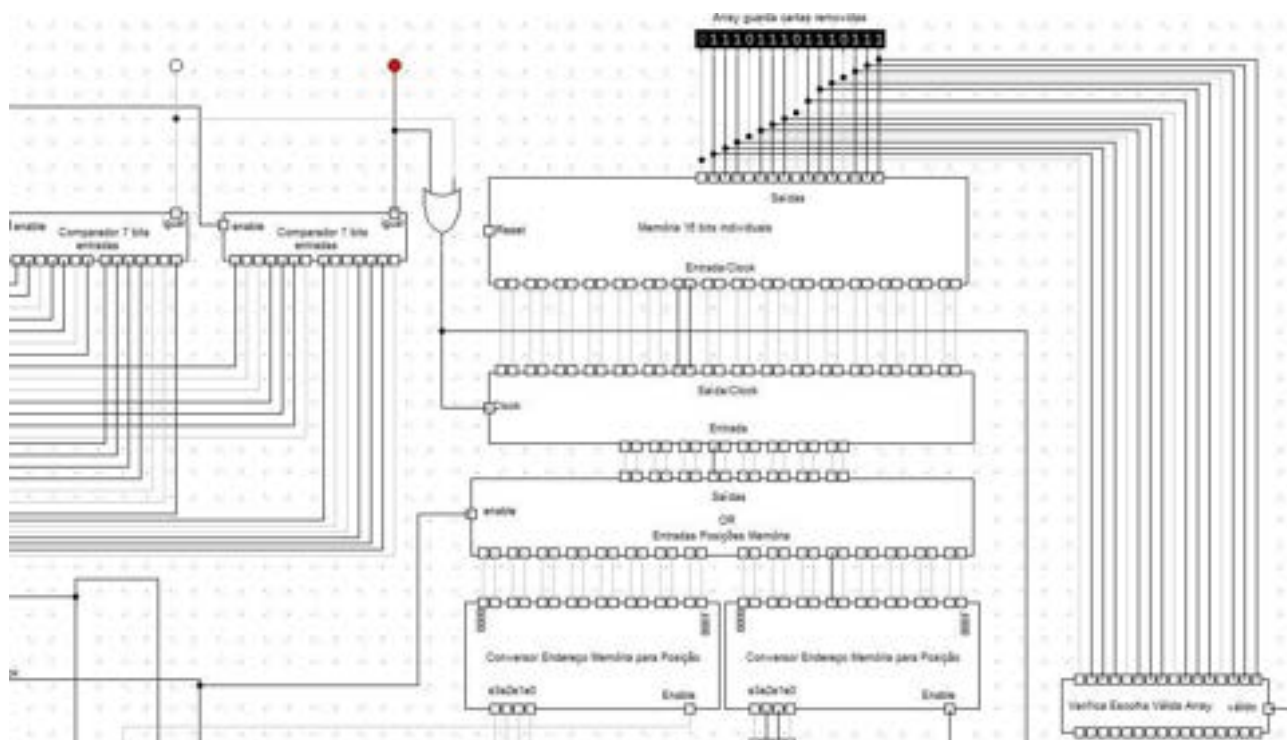
**Figura 76 – Alínea D – Transição**



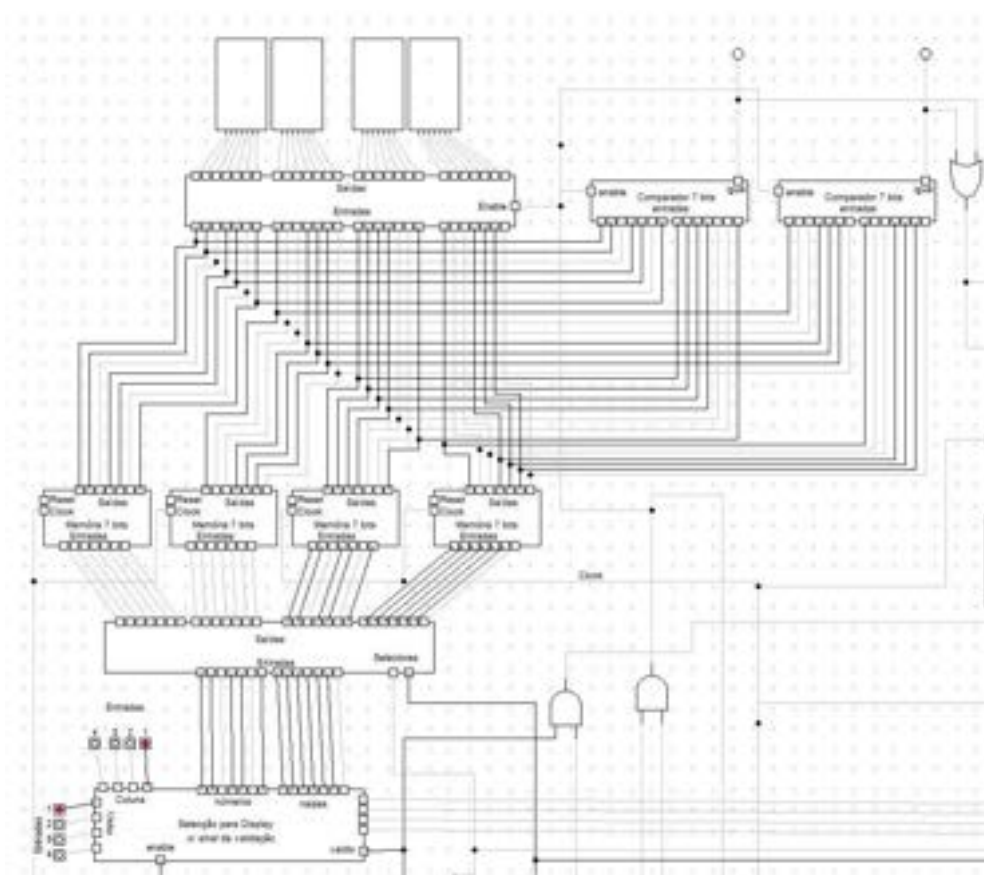
**Figura 77 – Alínea D – Teste - Seleção posição 22 – Carta 2C**



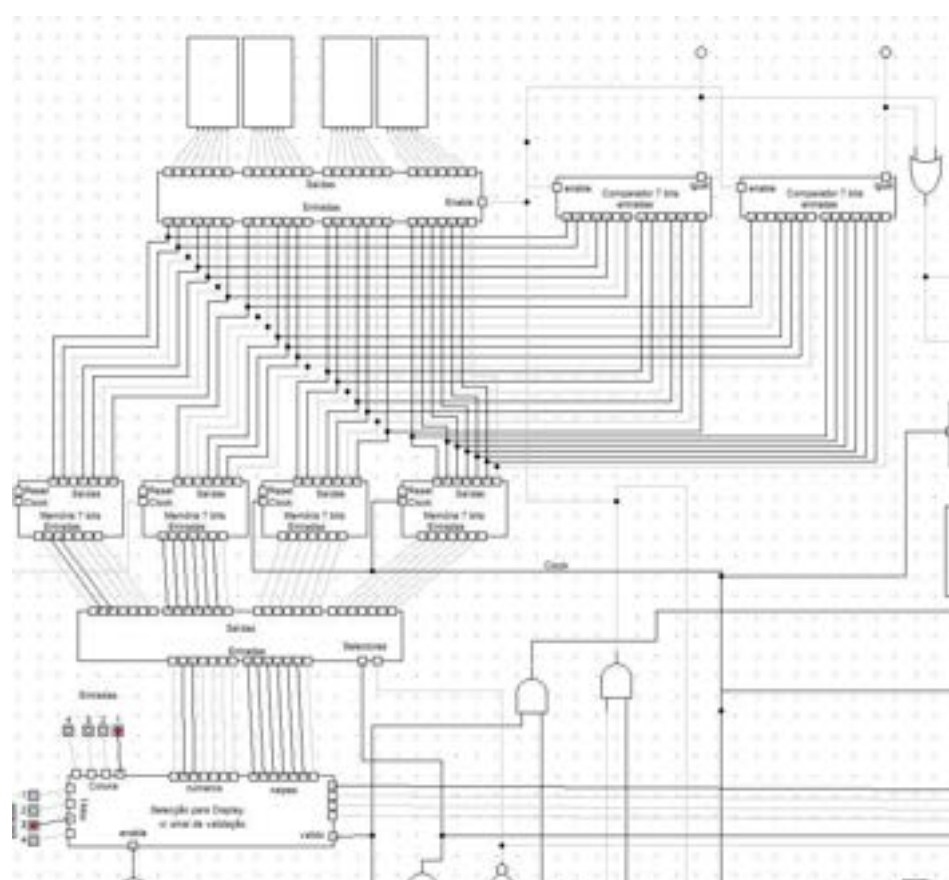
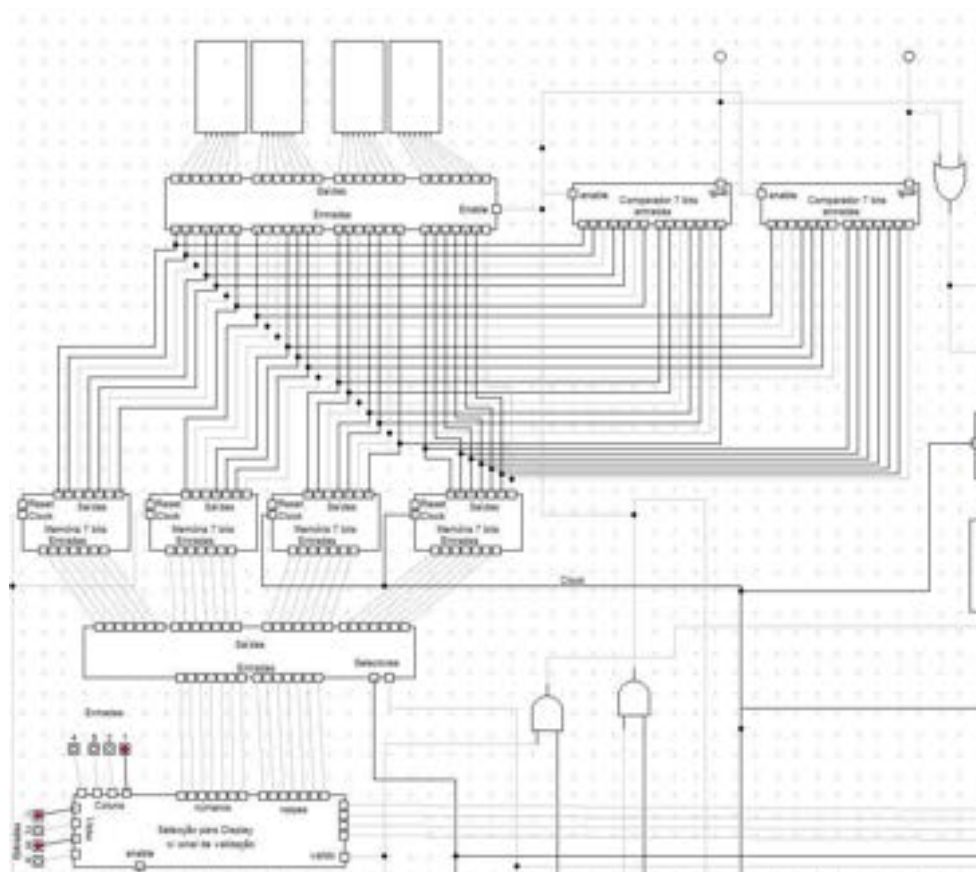
**Figura 78 – Alínea D – Teste – Display de Cartas**



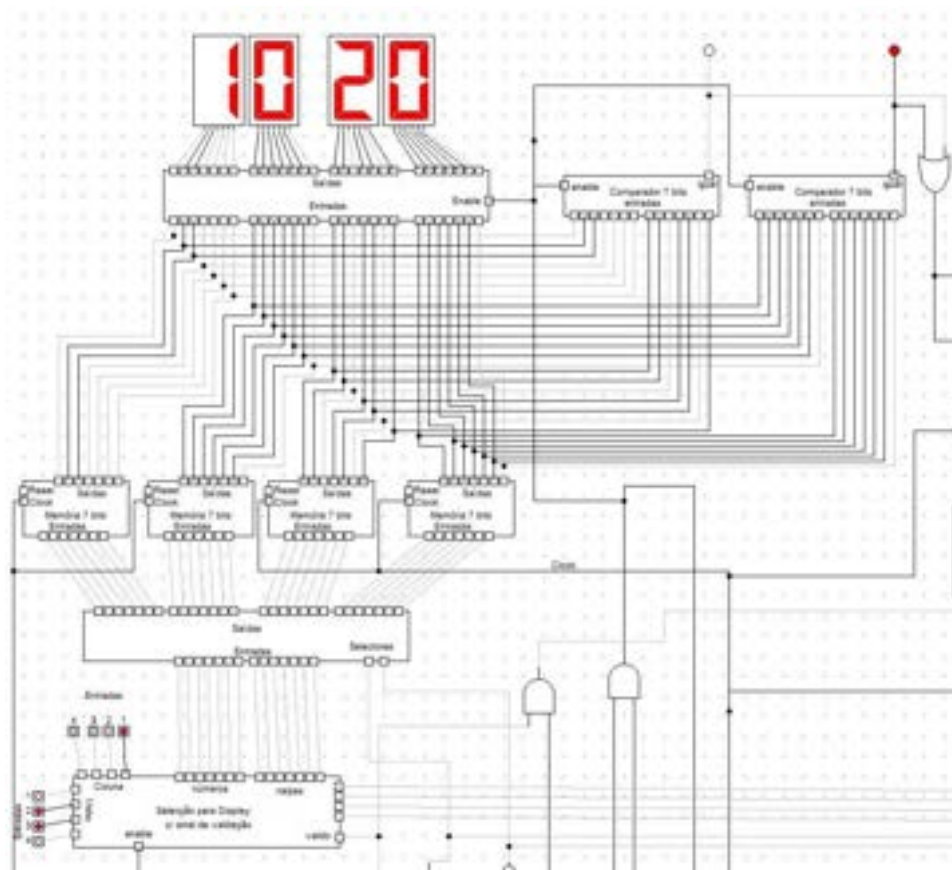
**Figura 79 – Alínea D – Teste – Guardar no Array**



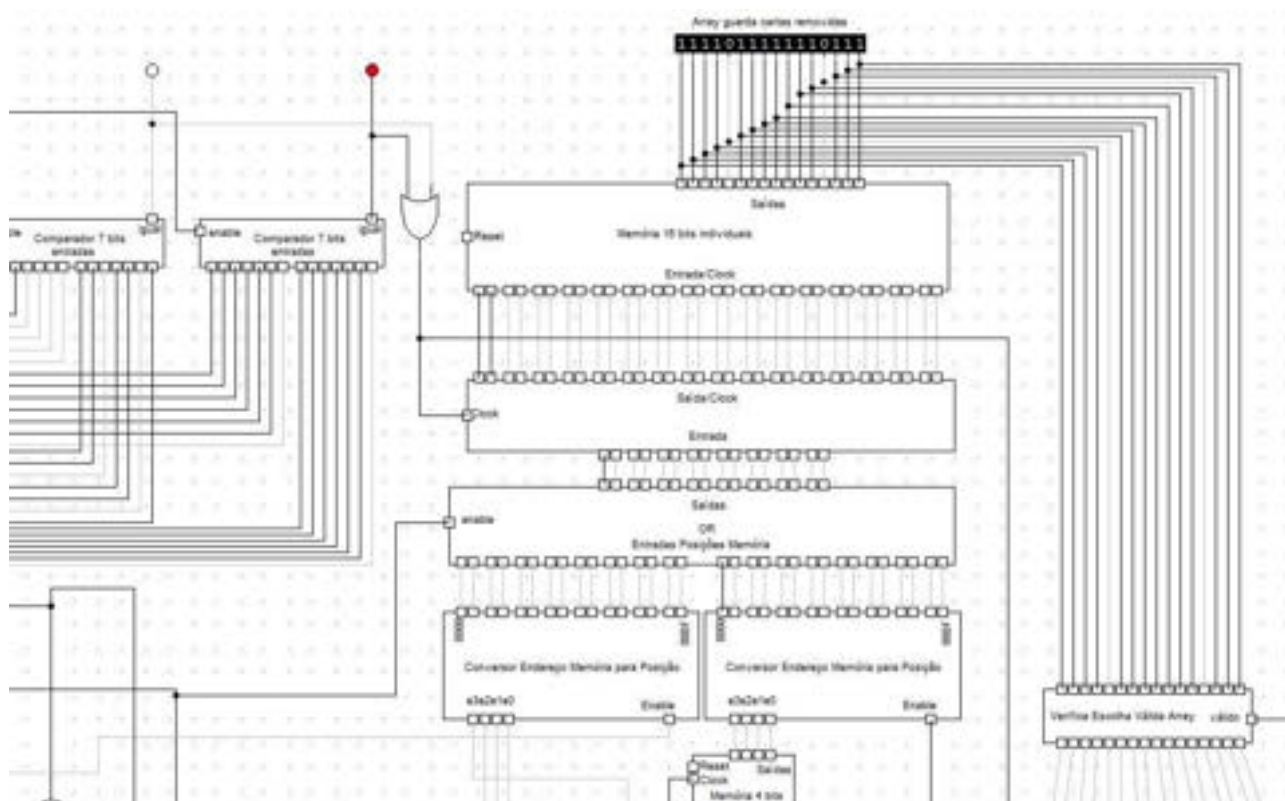
**Figura 80 – Alínea D – Teste - Seleção posição 11 – Carta 20**





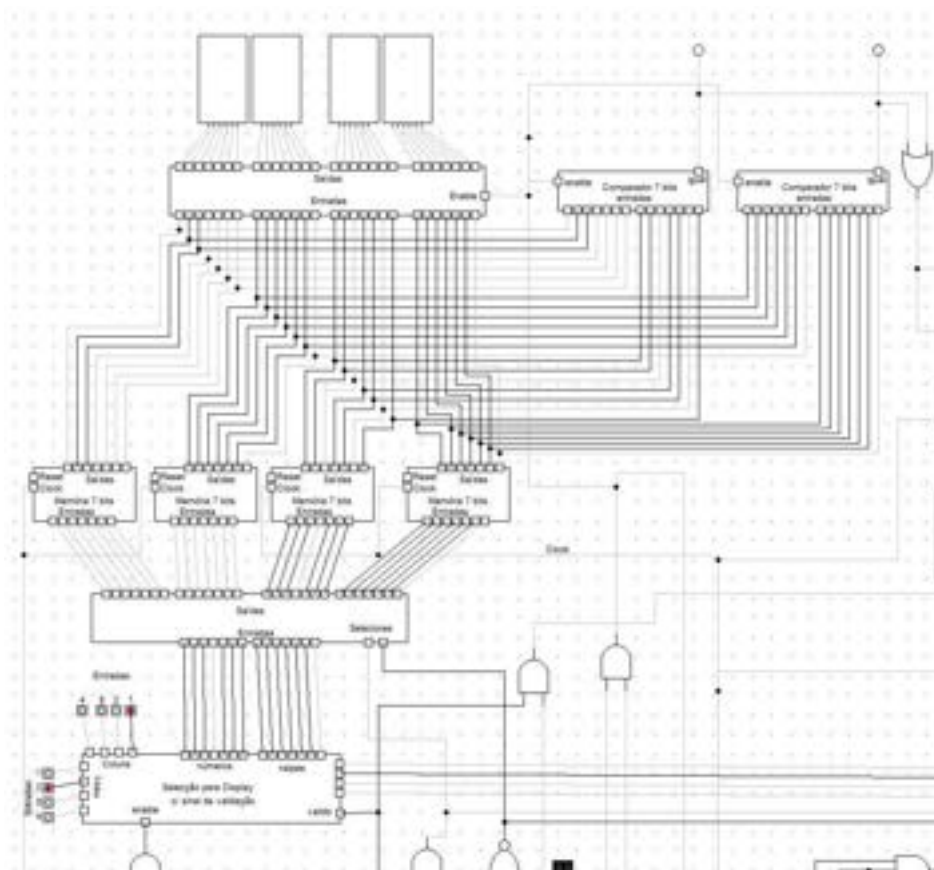


**Figura 83 – Alínea D – Teste – Display de Cartas**

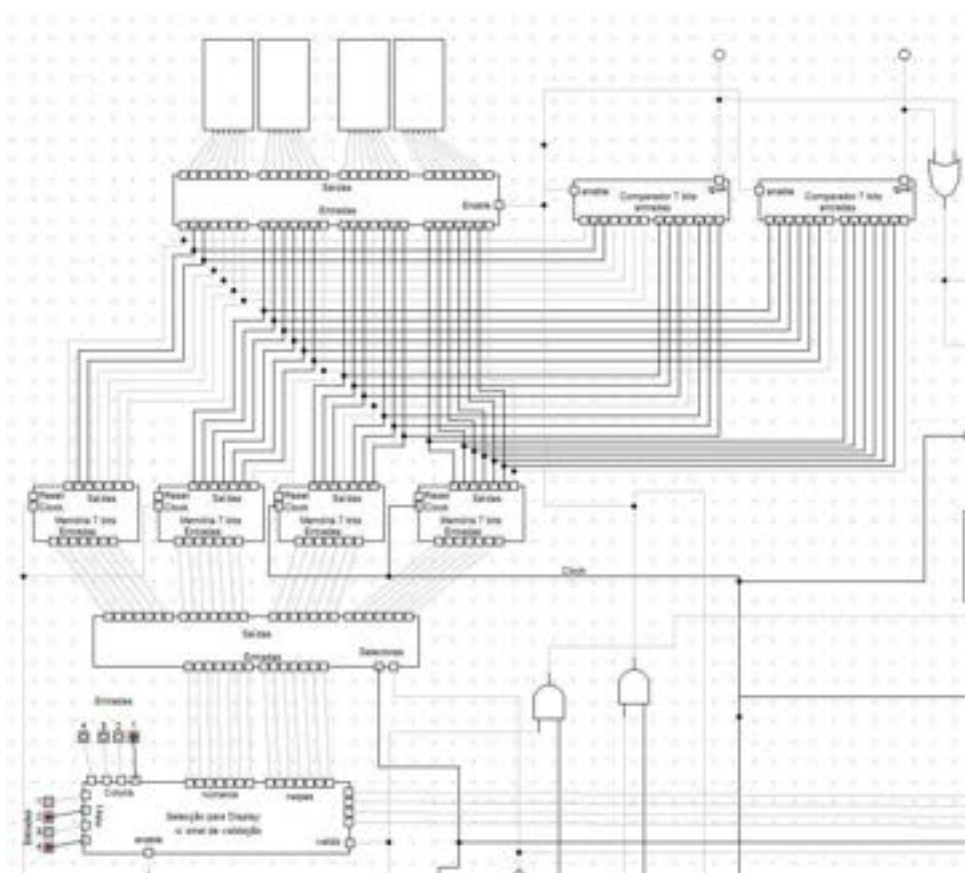


**Figura 84 – Alínea D – Teste – Guardar no Array**

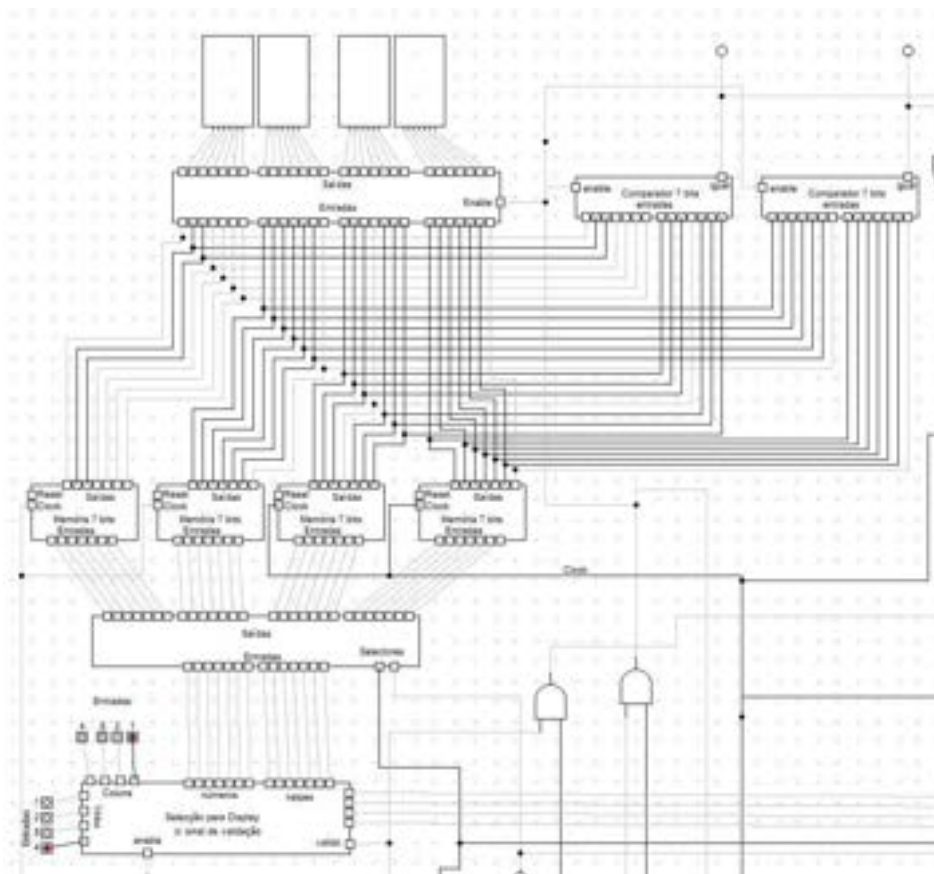




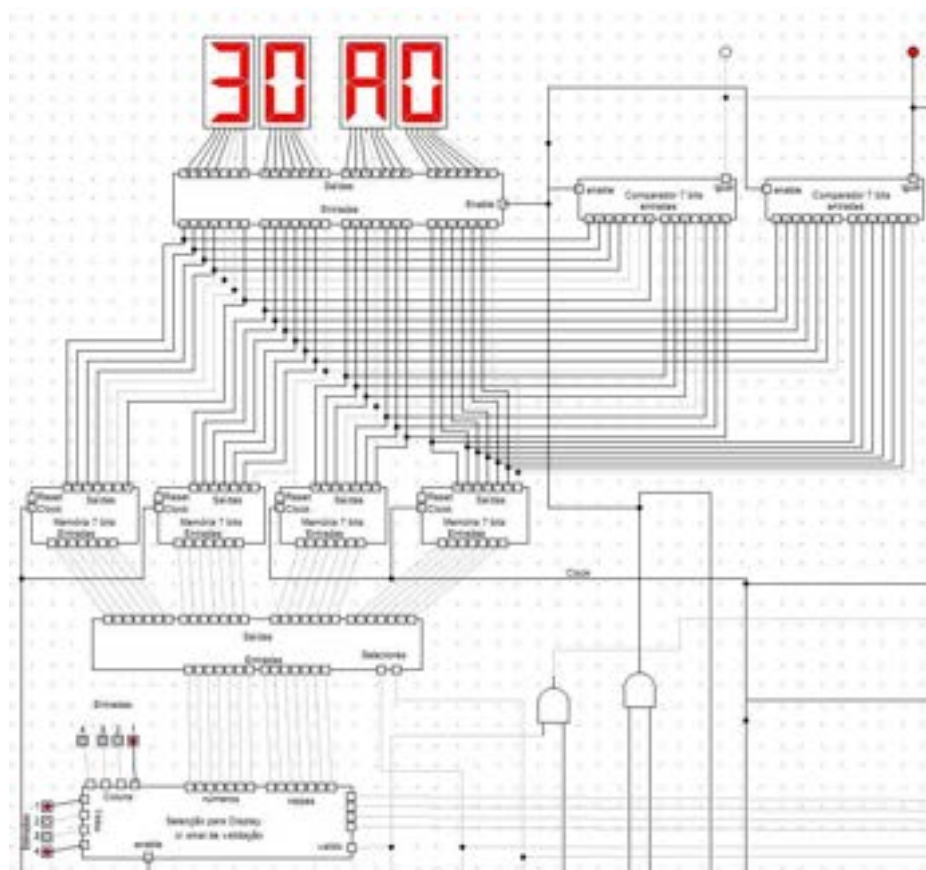
**Figura 85 – Alínea D – Teste - Seleção posição 21 – Carta AO**



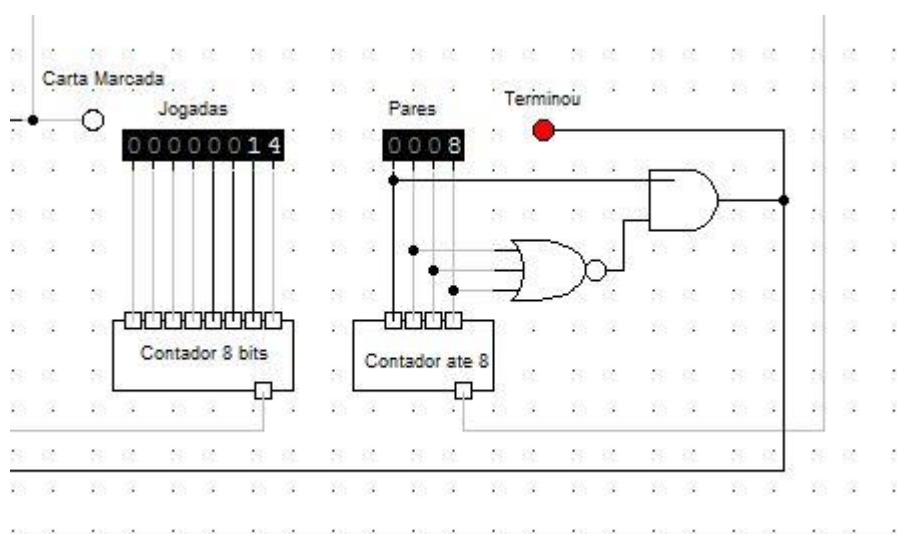
**Figura 86 – Alínea D – Transição**



**Figura 87 – Alínea D – Teste - Seleção posição 41 – Carta 30**



**Figura 88 – Alínea D – Teste – Display de Cartas**



**Figura 89 – Alínea D – Teste – Display de Contadores e Final do Jogo**