Neander - VHDL

Ronaldo Dall'Agnol Veiga

UFRGS - Instituto de Informática
Sistemas Digitais
Profa. Dra. Fernanda Gusmão de Lima Kastensmidt

Organização do Neander

Características

- Largura de dados e endereços de 8 bits
- Dados representados em complemento de dois
- 1 acumulador de 8 bits (AC)
- 1 apontador de programa de 8 bits (PC)
- 1 registrador de estado com 2 códigos de condição: negativo e zero
- Modo direto de endereçamento

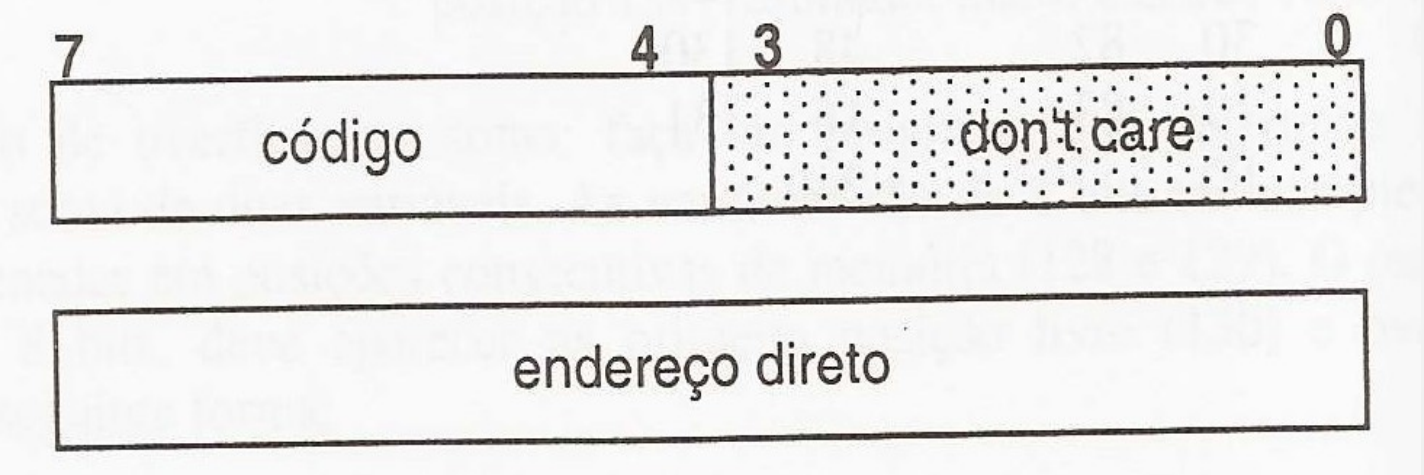
Conjunto de instruções

11 instruções codificadas através dos quatro bits mais significativos da palavra que contém o código da instrução

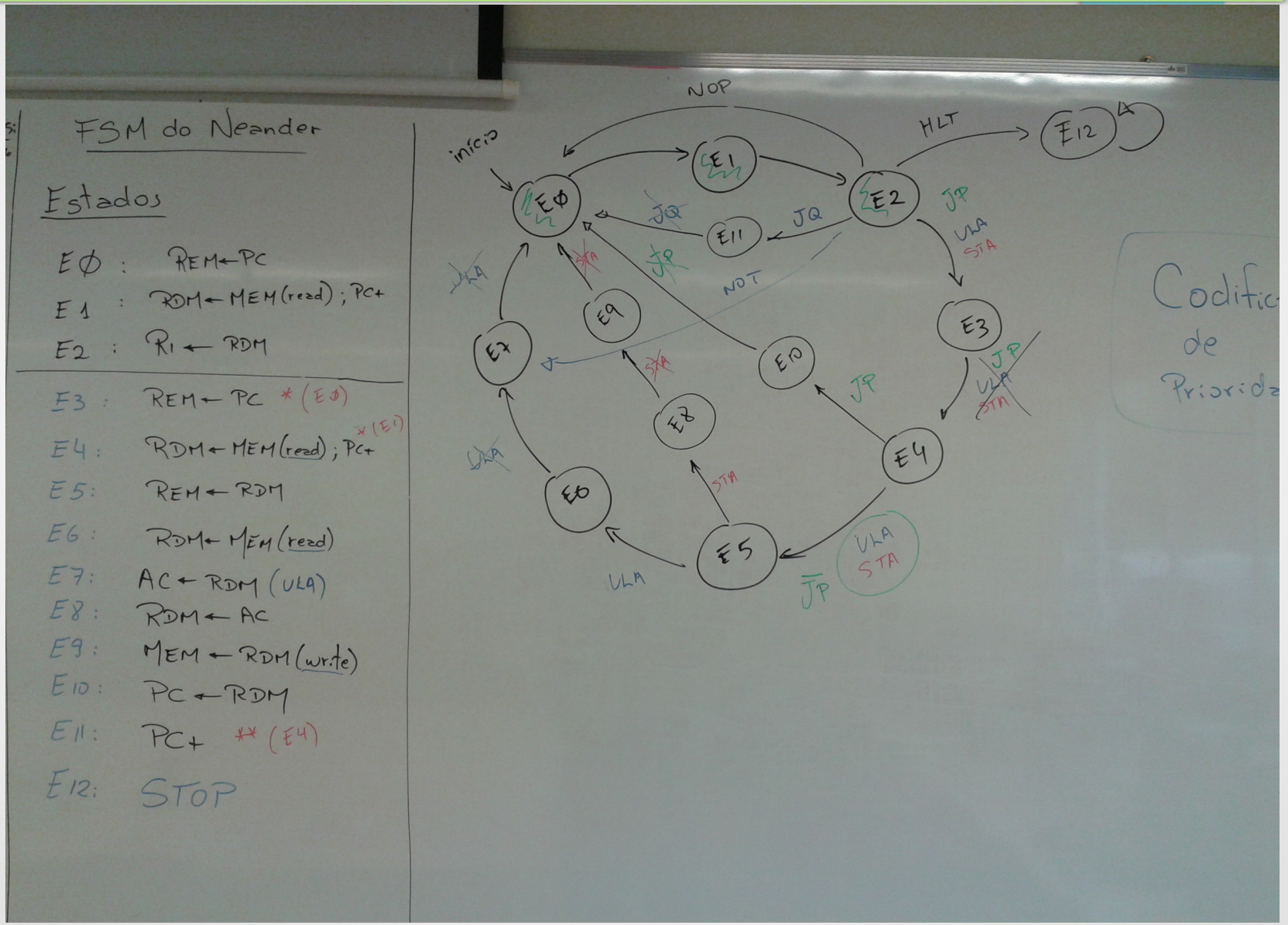
Código	Instrução	Comentário
0000	NOP	nenhuma operação
0001	STA end	armazena acumulador - (store)
0010	LDA end	carrega acumulador - (load)
0011	ADD end	soma
0100	OR end	"ou" lógico
0101	AND end	"e" lógico
0110	NOT	inverte (complementa) acumulador
1000	JMP end	desvio incondicional - (jump)
1001	JN end	desvio condicional - (jump on negative)
1010	JZ end	desvio condicional - (jump on zero)
1111	HLT	término de execução - (halt)

Formato das Instruções

Formadas por um ou dois bytes, ou seja, ocupam uma ou duas posições na memória.



Unidade de Controle



Fluxo de Dados

Código	Instrução	Execução
0000	NOP	nenhuma operação
0001	STA end	MEM(end) ← AC
0010	LDA end	AC← MEM(end)
0011	ADD end	AC← MEM(end) + AC
0100	OR end	AC← MEM(end) OR AC
0101	AND end	AC← MEM(end) AND AC
0110	NOT	AC← NOT AC
1000	JMP end	PC← end
1001	JN end	IF N=1 THEN PC ← end
1010	JZ end	IF Z=1 THEN PC ← end
1111	HLT	término de execução - (halt)

Exemplo de Programação

```
 0   32 128   LDA 128
 2   48 129   ADD 129
 4   64 130   OR  130
 6   80 129   AND 129
 8   96       NOT
 9  128  12   JMP 12
11  240       HLT
12  144  15   JN  15
14  240       HLT
15   48 131   ADD 131
17  160  20   JZ  20
19  240       HLT
20   32 130   LDA 130
22  144  30   JN  30
24   48 131   ADD 131
26  160  30   JZ  30
28   48 131   ADD 131
30  240       HLT
```

Sinais de Controle

Todos os sinais de controle da tabela 3 são gerados, nos instantes de tempo adequados, pela Unidade de Controle.

Transferência	Sinais de controle
REM ← PC	sel=0, carga REM
PC ← PC + 1	incrementa PC
RI ← RDM	carga RI
REM ← RDM	sel =1, carga REM
RDM ← AC	carga RDM
AC ← RDM; Atualiza N e Z	UAL(Y), carga AC, carga NZ
AC ← AC + RDM; Atualiza N e Z	UAL(ADD), carga AC, carga NZ
AC ← AC or RDM; Atualiza N e Z	UAL(OR), carga AC, carga NZ
AC ← AC and RDM; Atualiza N e Z	UAL(AND), carga AC, carga NZ
AC ← not(AC); Atualiza N e Z	UAL(NOT), carga AC, carga NZ
PC ← RDM	carga PC
RDM ← MEM(REM)	Read
MEM(REM) ← RDM	Write

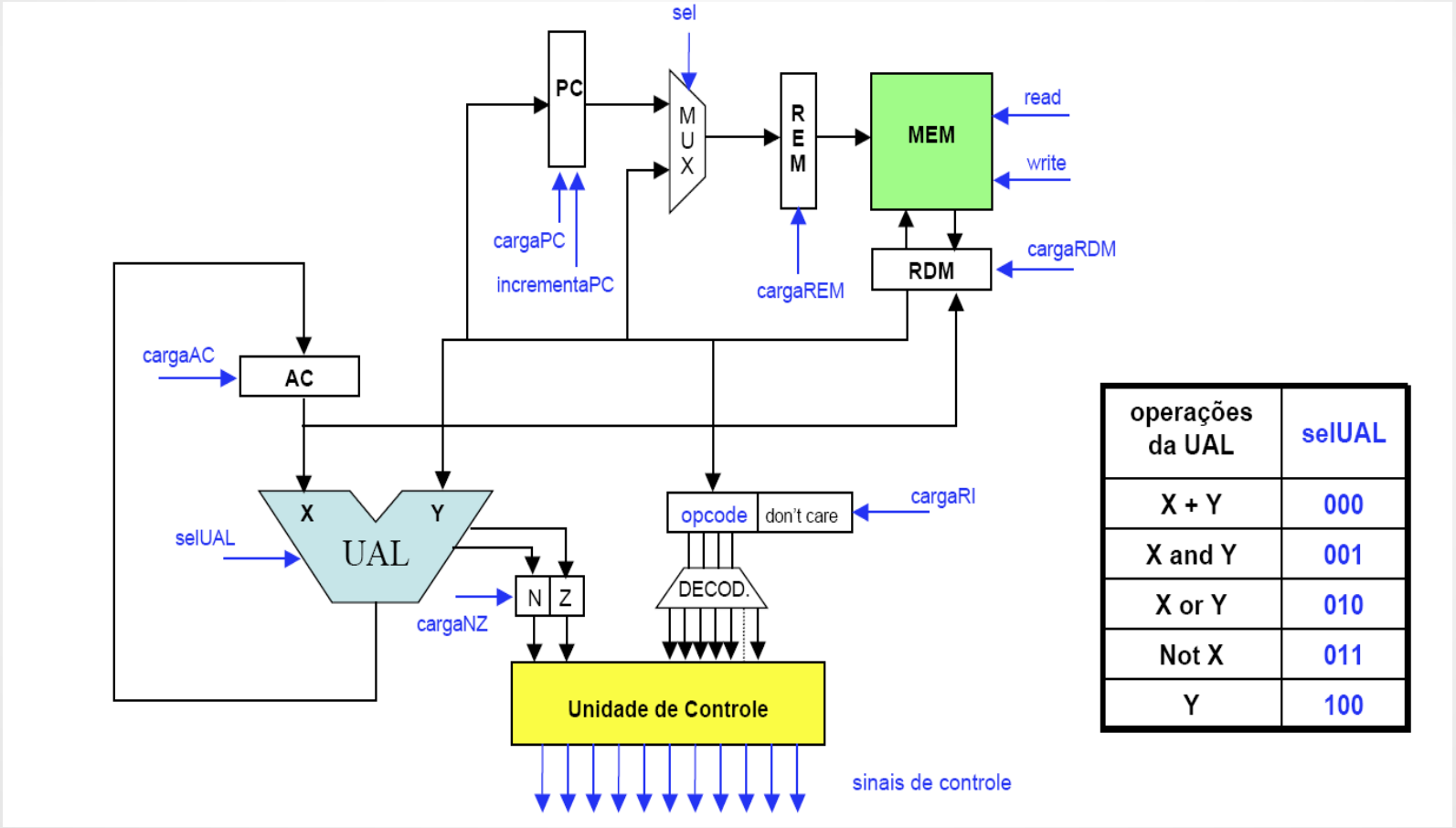
Sinais de Controle

tempo	STA	LDA	ADD	OR	AND	NOT
t0	sel=0, carga REM	sel=0, carga REM	sel=0, carga REM	sel=0, carga REM	sel=0, carga REM	sel=0, carga REM
t1	Read, incrementa PC	Read, incrementa PC	Read, incrementa PC	Read, incrementa PC	Read, incrementa PC	Read, incrementa PC
t2	carga RI	carga RI	carga RI	carga RI	carga RI	carga RI
t3	sel=0, carga REM	sel=0, carga REM	sel=0, carga REM	sel=0, carga REM	sel=0, carga REM	UAL(NOT), carga AC, carga NZ, goto t0
t4	Read, incrementa PC	Read, incrementa PC	Read, incrementa PC	Read, incrementa PC	Read, incrementa PC	
t5	sel=1, carga REM	sel=1, carga REM	sel=1, carga REM	sel=1, carga REM	sel=1, carga REM	
t6	carga RDM	Read	Read	Read	Read	
t7	Write, goto t0	UAL(Y), carga AC, carga NZ, goto t0	UAL(ADD), carga AC, carga NZ, goto t0	UAL(OR), carga AC, carga NZ, goto t0	UAL(AND, carga AC, carga NZ, goto t0	

Sinais de Controle

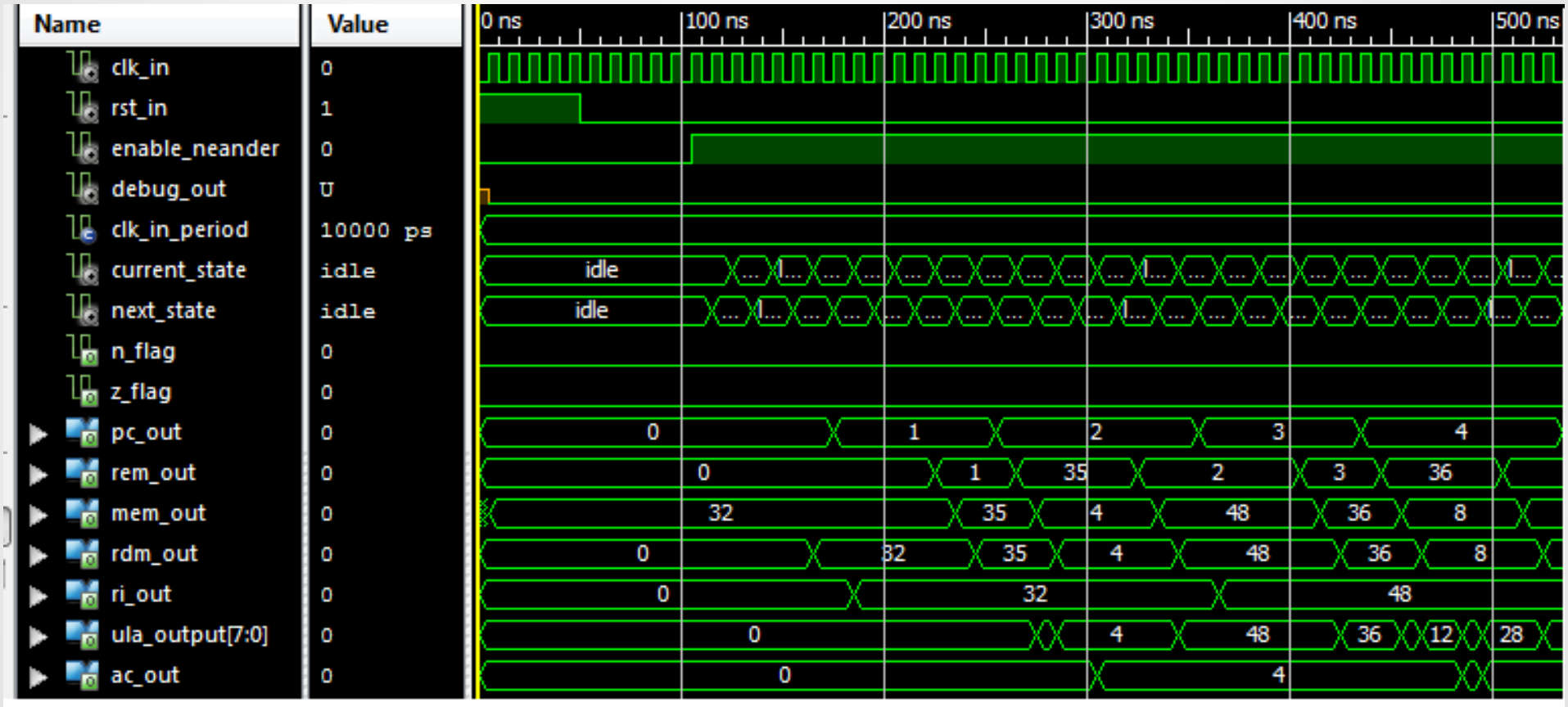
tempo	JMP	JN, N=1	JN, N=0	JZ, Z=1	JZ, Z=0	NOP	HLT
t0	sel=0, carga REM	sel=0, carga REM	sel=0, carga REM	sel=0, carga REM	sel=0, carga REM	sel=0, carga REM	sel=0, carga REM
t1	Read, increment a PC	Read, increment a PC	Read, increment a PC	Read, increment a PC	Read, increment a PC	Read, incrementa PC	Read, incrementa PC
t2	carga RI	carga RI	carga RI	carga RI	carga RI	carga RI	carga RI
t3	sel=0, carga REM	sel=0, carga REM	increment a PC, goto t0	sel=0, carga REM	increment a PC, goto t0	goto t0	Halt
t4	Read	Read		Read			
t5	carga PC, goto t0	carga PC, goto t0		carga PC, goto t0			
t6							
t7							

Visão geral do Neander

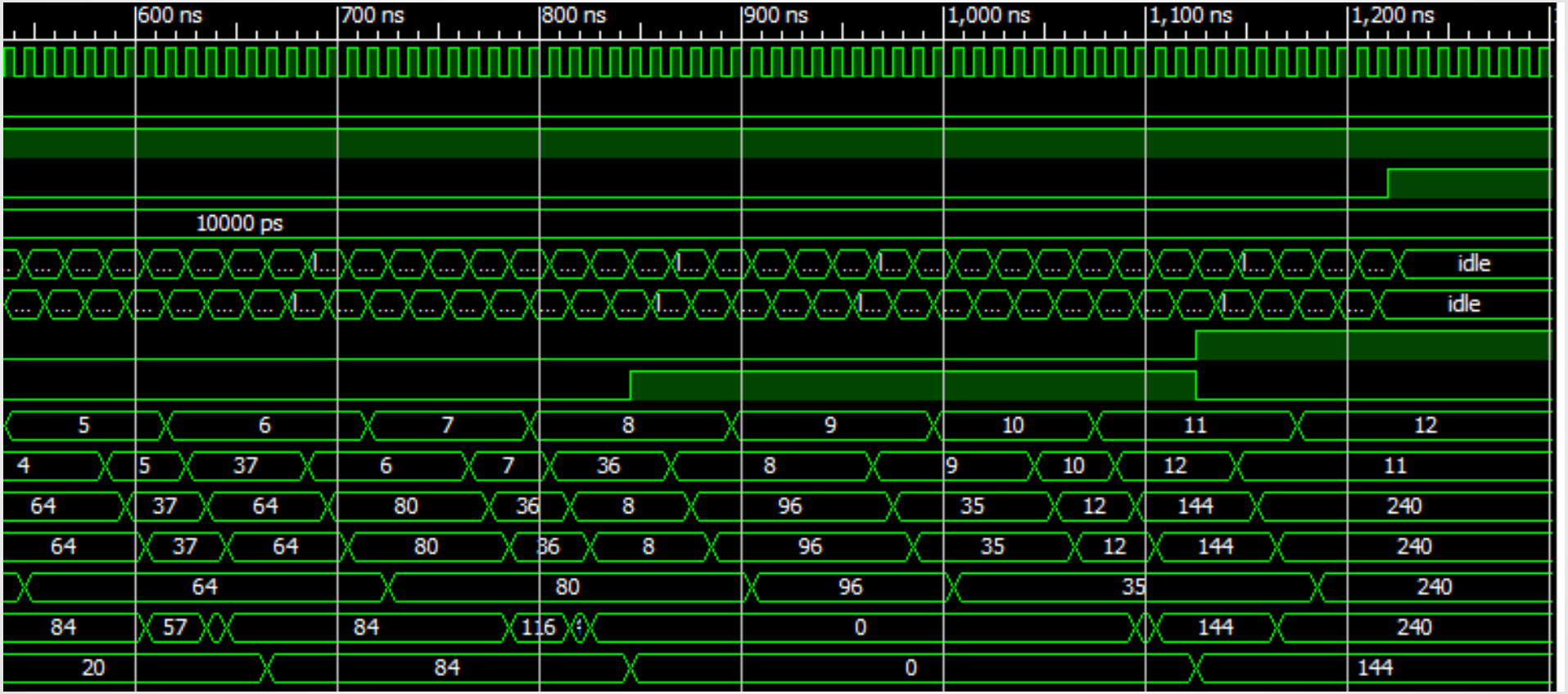


operações da UAL	selUAL
X + Y	000
X and Y	001
X or Y	010
Not X	011
Y	100

Simulação


Name
Value
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
clk_in
0
rst_in
1
enable_neander
0
debug_out
U
clk_in_period
10000 ps
current_state
idle
next_state
idle
n_flag
0
z_flag
0
pc_out
0
rem_out
0
mem_out
0
rdm_out
0
ri_out
0
ula_output[7:0]
0
ac_out
0

Simulação


600 ns
700 ns
800 ns
900 ns
1,000 ns
1,100 ns
1,200 ns
10000 ps
idle
idle
5 6 7 8 9 10 11 12
4 5 37 6 7 36 8 9 10 12 11
64 37 64 80 36 8 96 35 12 144 240
64 37 64 80 36 8 96 35 12 144 240
64 80 96 35 240
84 57 84 116 0 144 240
20 84 0 144