Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Digitaltechnik

Praktikum 1

Robert Palm
Florian Heuer

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Inhaltsverzeichnis

Vorwort

In der folgenden Dokumentation werden die Ergebnisse aus dem 1. Praktikum in Digitaltechnik unter Leitung von Prof. Dr. Schwarz zusammen getragen. Die genaue Aufgabenstellung ist dem Aufgabenblatt im Anhang zu entnehmen.

1. Paritätsbitgenerator

1.1 Warheitstabellen

1.1.1 2 Bit-Eingänge

Dezimal	A	B	P_ODD(AB)
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0

Dezimal	C	D	P_ODD(CD)
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0

Dezimal	P_ODD(AB)	P_ODD(CD)	P_ODD
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0

1.1.1 4 Bit-Eingänge

Dezimal	A	B	C	D	P_ODD
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0

1.2 Bool‘sche Algebra

1.2.1 DNF

$$(\neg A \wedge \neg B \wedge \neg C \wedge D)_1 \vee (\neg A \wedge \neg B \wedge C \wedge \neg D)_2 \vee (\neg A \wedge B \wedge \neg C \wedge \neg D)_4 \vee (\neg A \wedge B \wedge C \wedge D)_7$$
$$(A \wedge \neg B \wedge \neg C \wedge \neg D)_8 \vee (A \wedge \neg B \wedge C \wedge D)_{11} \vee (A \wedge B \wedge \neg C \wedge D)_{13} \vee (A \wedge B \wedge C \wedge \neg D)_{14}$$

1.2.2 Antivalenzdarstellung

A B C D

1.3 Umsetzung in VHDL

Der VHDL und do-File Code ist im Anhang zu finden. (siehe Seite 7)

1.4 Funktionsblockdiagramm

Das Funktionsblogdiagramm ist im Anhang zu finden. (siehe Seite 6)

1.5 Simulationsergebnis

Das Simulationsergebnis ist im Anhang zu finden. (siehe Seite 8)

1.6 Aufbau mit TTL-Gattern

Der Aufbau ist im Anhang zu finden. (siehe Seite)

2. Paritätschecker

2.1 Umsetzung in VHDL

Der VHDL und do-File Code ist im Anhang zu finden. (siehe Seite 9)

2.2 Simulationsergebnis

Der Simulationsergebnis ist im Anhang zu finden. (siehe Seite 10 & 11)

3. Kombination von Generator und Checker

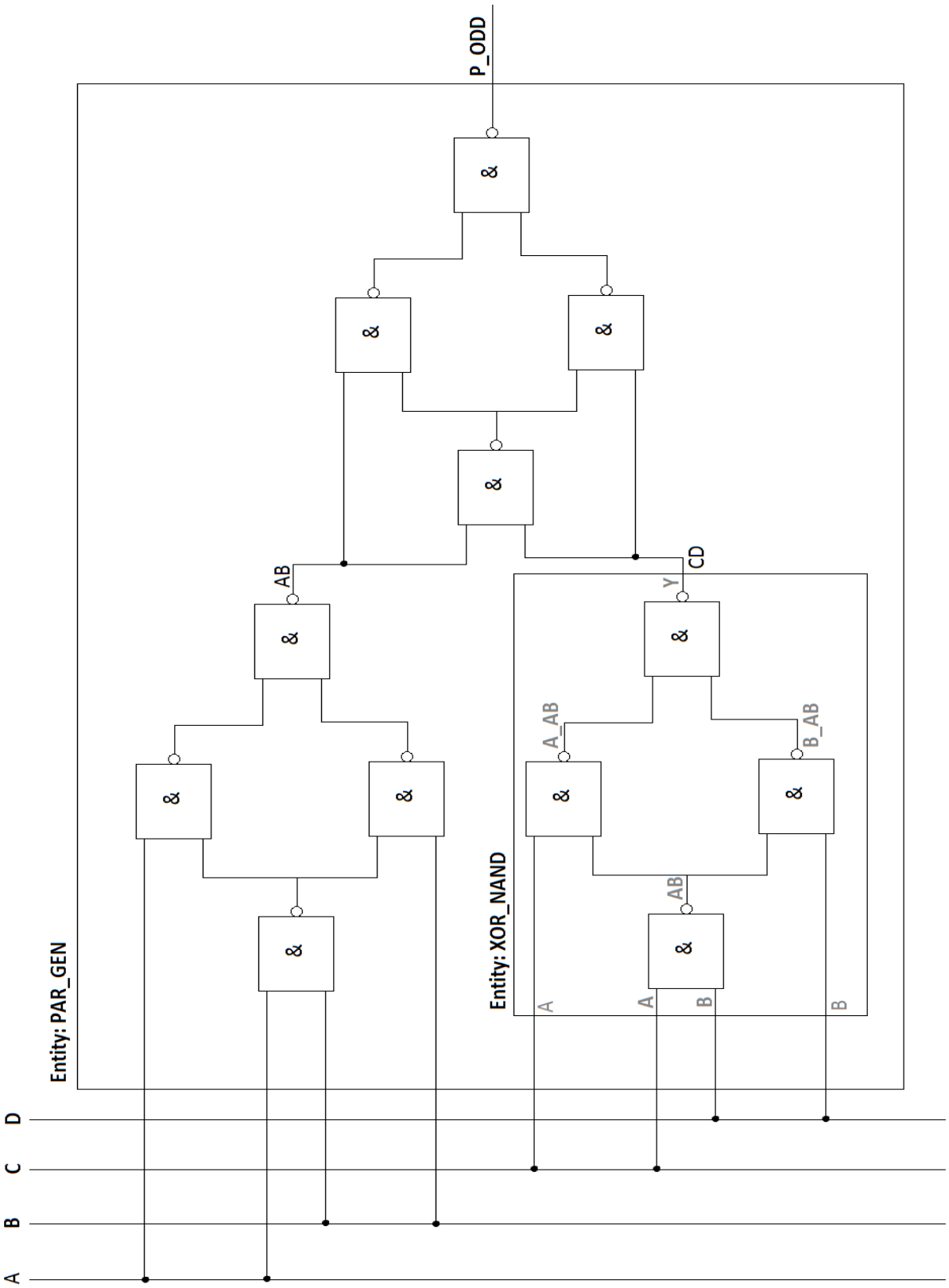
3.1 Umsetzung in VHDL

Der VHDL Code ist im Anhang zu finden. (siehe Seite)

3.2 Simulationsergebnis

Der Simulationsergebnis ist im Anhang zu finden. (siehe Seite)

4. Anhang


P_ODD
&
AB
CD
Y
A_AB
B_AB
AB
Entity: PAR_GEN
Entity: XOR_NAND
A
B
C
D

C:/Modeltech_pe_edu_10.4a/examples/ParitaetsbitGenerator.vhd (/par_gen) - Defau

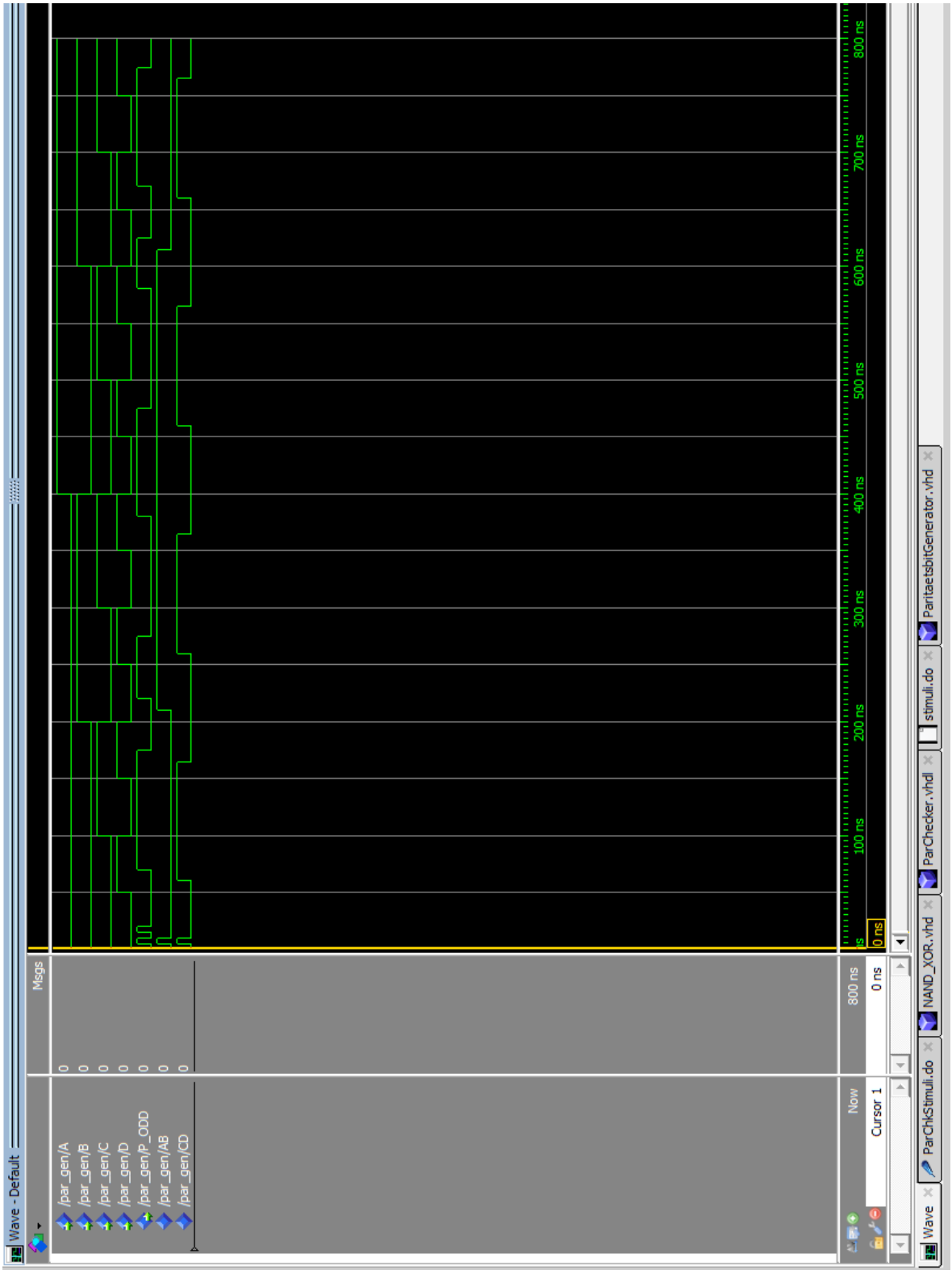
```
Ln#
 1  entity PAR_GEN is
 2          port( A, B, C, D : in bit ;
 3                P_ODD : out bit );
 4  end PAR_GEN;
 5
 6  architecture PAR_GEN_ARCH of PAR_GEN is
 7
 8          signal AB : bit;
 9          signal CD : bit;
10
11          begin
12
13          XOR_1 : entity work.XOR_NAND port map(
14                  A => A,
15                  B => B,
16                  Y => AB
17          );
18
19          XOR_2 : entity work.XOR_NAND port map(
20                  A => C,
21                  B => D,
22                  Y => CD
23          );
24
25          XOR_3 : entity work.XOR_NAND port map(
26                  A => AB,
27                  B => CD,
28                  Y => P_ODD
29          );
30
31  end PAR_GEN_ARCH;
```

C:/Modeltech_pe_edu_10.4a/examples/NAND_XOR.vhd (/par_checker/PAR_

```
Ln#
 1  entity XOR_NAND is
 2          port( A, B : in bit ; Y : out bit );
 3  end XOR_NAND;
 4
 5  architecture ARCH of XOR_NAND is
 6
 7          signal AB: bit;
 8          signal A_AB: bit;
 9          signal B_AB: bit;
10
11          begin
12
13          AB <= A nand B after 5ns;
14          A_AB <= AB nand A after 5ns;
15          B_AB <= AB nand B after 5ns;
16
17          Y <= A_AB nand B_AB after 5ns;
18
19  end ARCH;
```

C:/Modeltech_pe_edu_10.4a/examples/stimuli.do - Default

```
Ln#
 1  restart
 2  add wave *
 3  force D 0 0, 1 50 ns, 0 100 ns, 1 150 ns, 0 200 ns, 1 250 ns, 0 300 ns, 1 350 ns, 0 400 ns, 1 450 ns, 0 500 ns, 1 550 ns, 0 600 ns, 1 650 ns, 0 700 ns, 1 750 ns
 4  force C 0 0, 1 100 ns, 0 200 ns, 1 300 ns, 0 400 ns, 1 500 ns, 0 600 ns, 1 700 ns
 5  force B 0 0, 1 200 ns, 0 400 ns, 1 600 ns
 6  force A 0 0, 1 400 ns
 7
 8  run 800 ns
```

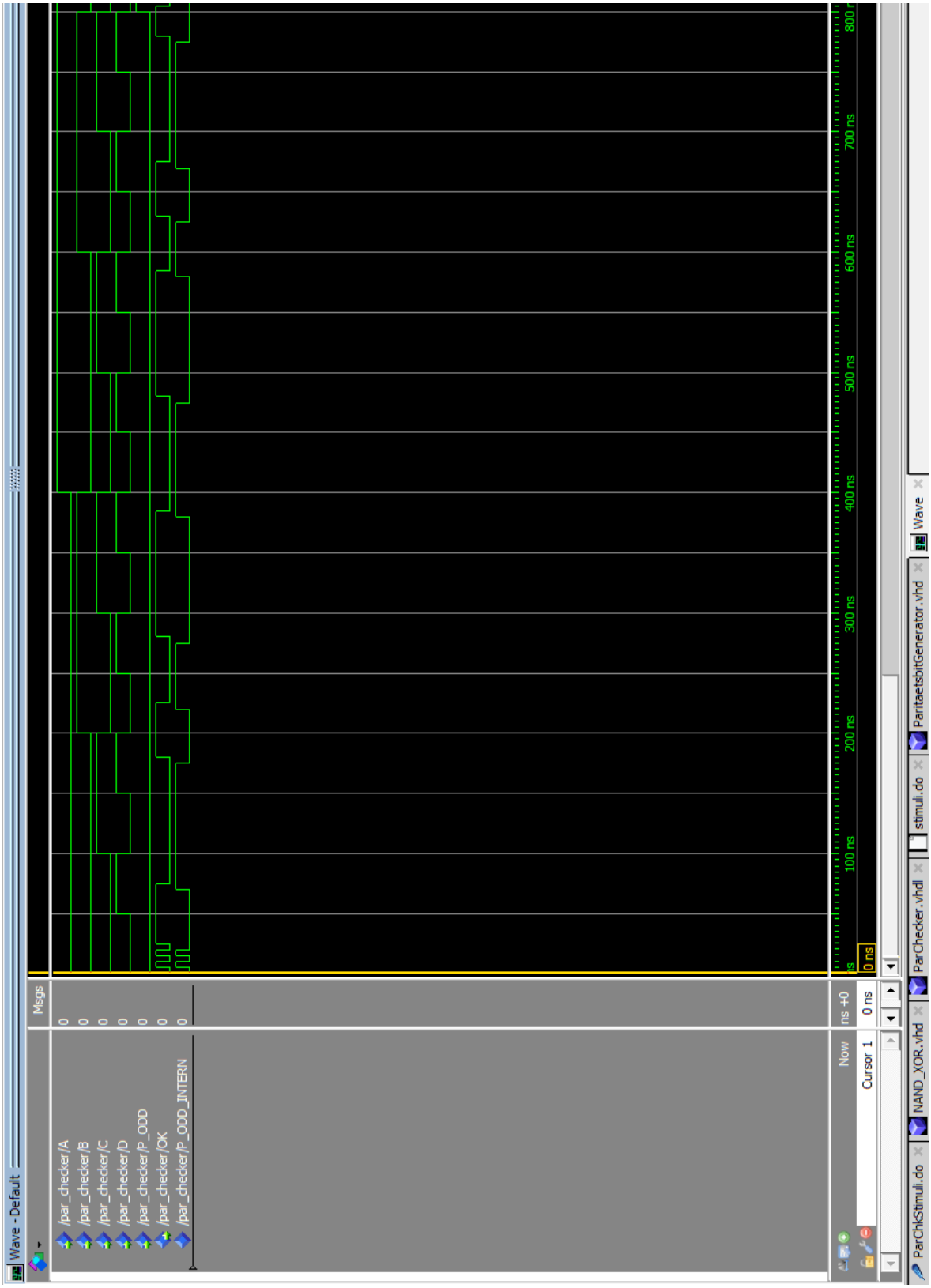

Wave - Default
Msgs
/par_gen/A
/par_gen/B
/par_gen/C
/par_gen/D
/par_gen/P_ODD
/par_gen/AB
/par_gen/CD
Now
800 ns
Cursor 1
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
Wave
ParChkStimuli.do
NAND_XOR.vhd
ParChecker.vhdl
stimuli.do
ParitaetsbitGenerator.vhd

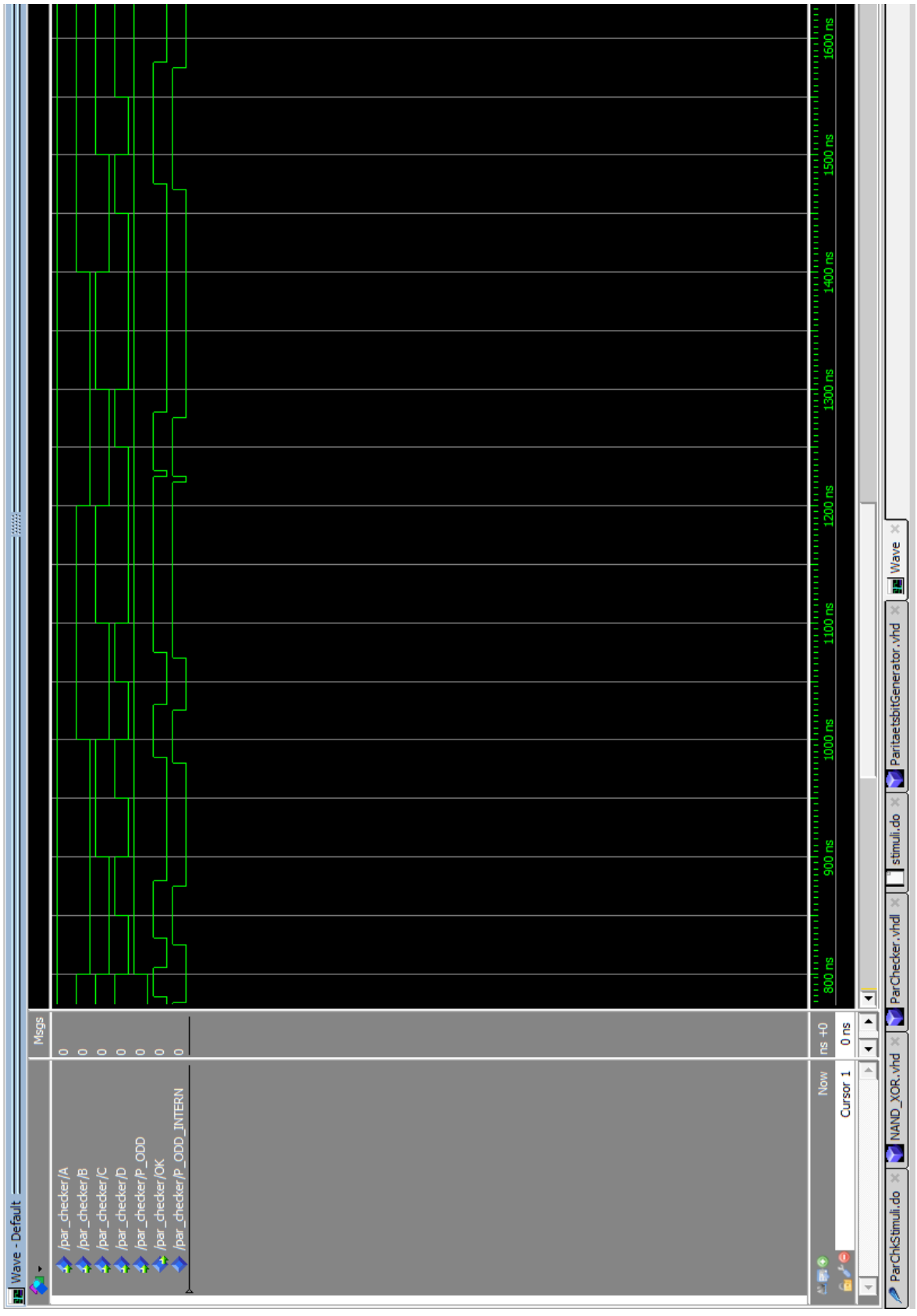
C:/Modeltech_pe_edu_10.4a/examples/ParChecker.vhdl - Default

```
Ln#
 1  entity PAR_CHECKER is
 2          port( A, B, C, D, P_ODD : in bit ; OK : out bit );
 3  end PAR_CHECKER;
 4
 5  architecture PAR_CHECKER_ARCH of PAR_CHECKER is
 6
 7          signal P_ODD_INTERN : bit;
 8
 9          begin
10
11          PAR_GEN : entity work.PAR_GEN port map(
12                  A => A,
13                  B => B,
14                  C => C,
15                  D => D,
16                  P_ODD => P_ODD_INTERN
17          );
18
19          OK <= P_ODD_INTERN xnor P_ODD after 5 ns;
20
21  end PAR_CHECKER_ARCH;
```

C:/Modeltech_pe_edu_10.4a/examples/ParChkStimuli.do - Default *

```
Ln#
 1  restart
 2  add wave *
 3  force D 0 0, 1 50 ns, 0 100 ns, 1 150 ns, 0 200 ns, 1 250 ns, 0 300 ns, 1 350 ns, 0 400 ns, 1 450 ns, 0 500 ns, 1 550 ns, 0 600 ns, 1 650 ns, 0 700 ns, 1 750 ns, 0 800 ns,
 4      1 850 ns, 0 900 ns, 1 950 ns, 0 1000 ns, 1 1050 ns, 0 1100 ns, 1 1250 ns, 0 1300 ns, 1 1450 ns, 0 1500 ns, 1 1550 ns
 5
 6  force C 0 0, 1 100 ns, 0 200 ns, 1 300 ns, 0 400 ns, 1 500 ns, 0 600 ns, 1 700 ns, 0 800 ns, 1 900 ns, 0 1000 ns, 1 1100 ns, 0 1200 ns, 1 1300 ns, 0 1400 ns, 1 1500 ns
 7  force B 0 0, 1 200 ns, 0 400 ns, 1 600 ns, 0 800 ns, 1 1000 ns, 0 1200 ns, 1 1400 ns
 8  force A 0 0, 1 400 ns, 1 800 ns, 1 1200 ns
 9  force P_ODD 0 0, 1 800 ns
10
11  run 1600 ns
```


Wave - Default
Msgs
/par_checker/A
/par_checker/B
/par_checker/C
/par_checker/D
/par_checker/P_ODD
/par_checker/OK
/par_checker/P_ODD_INTERN
Now
Cursor 1
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
ParChkStimuli.do
NAND_XOR.vhd
ParChecker.vhdl
stimuli.do
ParitaetsbitGenerator.vhd
Wave


Wave - Default
Msgs
/par_checker/A
/par_checker/B
/par_checker/C
/par_checker/D
/par_checker/P_ODD
/par_checker/OK
/par_checker/P_ODD_INTERN
Now
Cursor 1
800 ns
900 ns
1000 ns
1100 ns
1200 ns
1300 ns
1400 ns
1500 ns
1600 ns
ParChkStimuli.do
NAND_XOR.vhd
ParChecker.vhdl
stimuli.do
ParitaetsbitGenerator.vhd
Wave