

PROGRAMMABLE ASIC DESIGN SOFTWARE



Key concepts: There are five components of a programmable ASIC or FPGA :

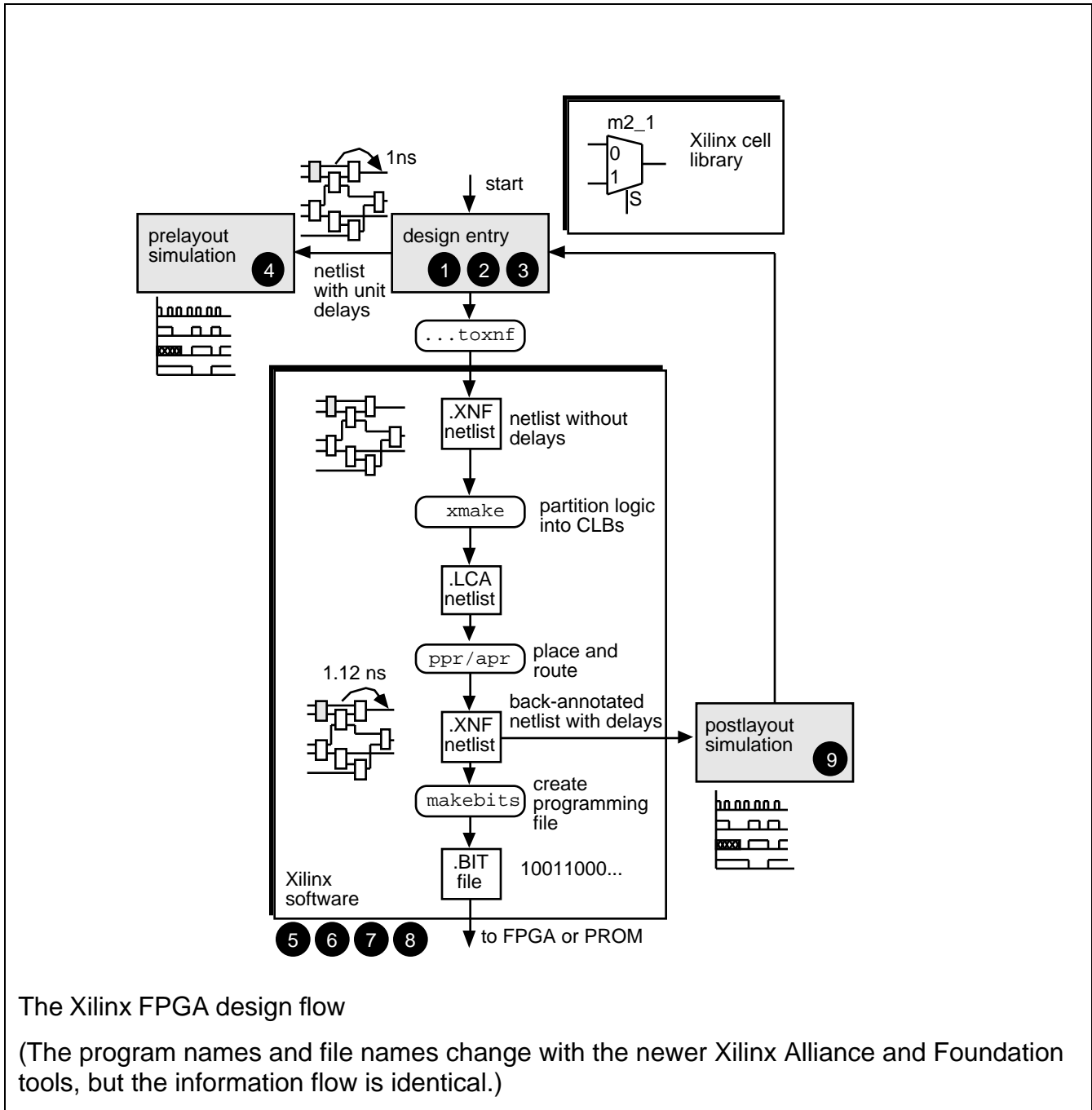
- (1) the programming technology
- (2) the basic logic cell
- (3) the I/O cell
- (4) the interconnect
- (5) the **design software** that allows you to program the ASIC

The design software is much more closely tied to the FPGA architecture than is the case for other types of ASICs

8.1 Design Systems

Keywords: design kits • original equipment manufacturer (OEM) • generic cell library • hardware description languages (HDLs) • ABEL (pronounced “able”) • CUPL (“cupple”) • PALASM (“pal-azzam”) • VHDL • Verilog • logic simulator • back-annotation • postlayout timing information • postlayout netlist (also called a back-annotated netlist) • postlayout timing simulation • timing-analysis • timing constraint • timing violation • forward-annotation

8.1.1 Xilinx



The Xilinx FPGA design flow

(The program names and file names change with the newer Xilinx Alliance and Foundation tools, but the information flow is identical.)

8.1.2 Actel

File types used by Actel design software (an example—these change often)

ADL	Main design netlist
IPF	Partial or complete pin assignment for the design
CRT	Net criticality
VALIDATED	Audit information
COB	List of macros removed from design
VLD	Information, warning, and error messages
PIN	Complete pin assignment for the design
DFR	Information about routability and I/O assignment quality
LOC	Placement of non-I/O macros, pin swapping, and freeway assignment
PLI	Feedback from placement step
SEG	Assignment of horizontal routing segments
STF	Back-annotation timing
RTI	Feedback from routing step
FUS	Fuse coordinates (column-track, row-track)
DEL	Delays for input pins, nets, and I/O modules
AVI	Fuse programming times and currents for last chip programmed

FPGA state-machine language (an example of “third-party” tools)**LOG/iC state-machine language****PALASM version**

```

*IDENTIFICATION
sequence detector
LOG/iC code
*X-NAMES
X; !input
*Y-NAMES
D; !output, D = 1 when three 1's
appear on X
*FLOW-TABLE
;State, X input, Y output, next
state
  S1,  X1,  Y0,  F2;
  S1,  X0,  Y0,  F1;
  S2,  X1,  Y0,  F3;
  S2,  X0,  Y0,  F1;
  S3,  X1,  Y0,  F4;
  S3,  X0,  Y0,  F1;
  S4,  X1,  Y1,  F4;
  S4,  X0,  Y0,  F1;
*STATE-ASSIGNMENT
BINARY;
*RUN-CONTROL
PROGFORMAT = P-EQUATIONS;
*END

```

```

TITLE sequence detector
CHIP MEALY USER
CLK Z QQ2 QQ1 X
EQUATIONS
Z = X * QQ2 * QQ1
QQ2 := X * QQ1 + X * QQ2
QQ1 := X * QQ2 + X * /QQ1

```

8.1.3 Altera

Altera uses a self-contained design system, **MAX+plus** (as well as an interface to EDIF for third-party schematic entry or logic synthesis).

- The interconnect scheme in Altera complex PLDs is nearly **deterministic**, simplifying the physical-design software as well as eliminating the need for back-annotation and a postlayout simulation.
- As Altera FPGAs become larger and more complex, some cases require signals to make more than one pass through the routing structures or travel large distances across the Altera **FastTrack interconnect**. It is possible to tell if this will be the case only by trying to place and route an Altera device.

8.2 Logic Synthesis

It is easier to write $A = B + C$ than to draw an FPGA schematic for a 32-bit adder at the gate level

Key concepts, facts, and terms: logic synthesis • logic minimization • **mapping** • fine-grain architecture • coarse-grain architecture • vendor independence • Synplicity • Synopsys FPGA Express • FPGA Compiler • Design Compiler • Exemplar • X-BLOX • LPM • IP cores

8.2.1 FPGA Synthesis

The VHDL code for a sequence detector

```

entity detector is port (X, CLK: in BIT; Z : out BIT); end;

architecture behave of detector is
  type states is (S1, S2, S3, S4);
  signal current, next: states;
begin
  combinational: process begin
    case current is
      when S1 =>
        if X = '1' then Z <= '0'; next <= S3; else Z <= '0'; next <=
S1; end if;
      when S2 =>
        if X = '1' then Z <= '0'; next <= S2; else Z <= '0'; next <=
S1; end if;
      when S3 =>
        if X = '1' then Z <= '0'; next <= S2; else Z <= '0'; next <=
S1; end if;
      when S4 =>
        if X = '1' then Z <= '1'; next <= S4; else Z <= '0'; next <=
S1; end if
    end case;
  end process
  sequential: process begin
    wait until CLK'event and CLK = '1'; current <= next ;
  end process;
end behave;

```

A Synopsys script

```


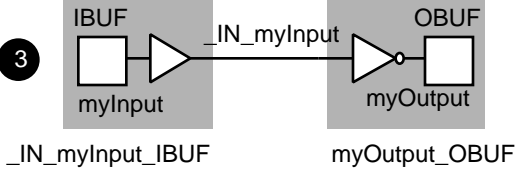
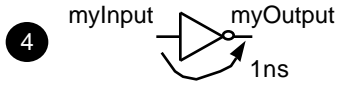
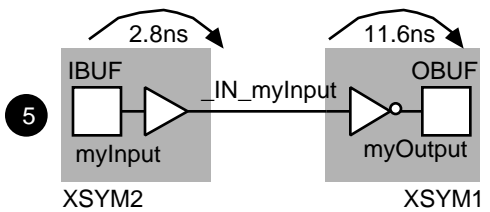
/design checking/
search_path = .
/use the TI cell libraries/
link_library = tpc10.db
target_library = tpc10.db
symbol_library = tpc10.sdb
read -f vhd1 detector.vhd
current_design = detector
write -n -f db -hierarchy -0
detector.db
check_design > detector.rpt

report_design > detector.rpt
/optimize for area/
max_area 0.0
compile
write -h -f db -o detector_opt.db
report -area -cell -timing >
detector.rpt
free -all
/write EDIF netlist/
write -h -f edif -0
exit

```

8.3 The Halfgate ASIC

8.3.1 Xilinx

Design flow for the Xilinx implementation of the halfgate ASIC	Design flow
Script (using Compass tools as an example)	Design flow
<pre> # halfgate.xilinx.inp shell setdef path working xc4000d xblox cmosch000x quit asic open [v]halfgate synthesize save [nls]halfgate_p quit fpga set tag xc4000 set opt area optimize [nls]halfgate_p quit qtv open [nls]halfgate_p trace critical print trace [txt]halfgate_p quit shell vuterm exec xnfmerge -p 4003PC84 halfgate_p > /dev/null exec xnfprep halfgate_p > /dev/null exec ppr halfgate_p > /dev/null exec makebits -w halfgate_p > /dev/null exec lca2xnf -g -v halfgate_p halfgate_b > /dev/null quit manager notice utility netlist open [xnf]halfgate_b save [nls]halfgate_b save [edf]halfgate_b quit qtv open [nls]halfgate_b trace critical print trace [txt]halfgate_b quit </pre>	<p>1 myOutput = ~myInput</p> <p>2 myInput  myOutput</p> <p>3 </p> <p>4 </p> <p>5 </p>

The Xilinx files for the halfgate ASIC**Verilog file** (halfgate.v)

```
module halfgate(myInput, myOutput); input myInput; output
myOutput; wire myOutput;
  assign myOutput = ~myInput;
endmodule
```

Preroute XNF file (halfgate_p.xnf)

```
LCANET, 5                                END
USER, FPGA-Optimizer, 4.1,              EXT, myInput, I,
Date:960710 , Option: Area              SYM,
PROG, FPGA-Optimizer, 4.1,              myOutput_obuf, OBUF, LIBVER=
"Lib=4000"                               2.0.0,
PART, 4010PG191                          PIN, I, I, _IN_myInput,,
PWR, 0, GND                               INV
PWR, 1, VCC                               PIN, O, O, myOutput,
SYM, _IN_myInput_IBUF, IBUF, LIB        END
VER = 2.0.0                               EXT, myOutput, O,
PIN, I, I, myInput,                       EOF
PIN, O, O, _IN_myInput,
```

LCA file (halfgate_p.lca)

```

;: halfgate_p.lca (4003PC84-      Editblk PAD61
4), makebits 5.2.0, Tue Jul 16   Base IO
20:09:43 1996                    Config INFF: I1: I2:I 0:
Version 2                          OUT: PAD: TRI:
Design 4003PC84 4 0                Endblk
Speed -4                            Editblk PAD1
Addnet PAD_myInput PAD61.I2        Base IO
PAD1.0                              Config INFF: I1: I2: 0:
Netdelay PAD_myInput PAD1.0        OUT:0:NOT PAD: TRI:
3.1                                Endblk
Program PAD_myInput {65G521}       Nameblk PAD61 myInput
{65G287} {65G50} {63G50}          Nameblk PAD1 myOutput
{52G50} {45G50}                  Intnet myOutput PAD
NProgram PAD_myInput              myOutput
col.B.long.3:PAD1.0               Intnet myInput PAD myInput
col.B.long.3:row.G.local.1        System FGG 0 VERS 2 !
col.B.long.3:row.M.local.5-s     System FGG 1 GD0 0 !
MB.
40.1.14 MB.40.1.35
row.M.local.5:PAD61.I2

```

Postroute XNF file (halfgate_b.xnf)

```

LCANET, 4                          EXT, myOutput, 0, 10
PROG, LCA2XNF, 5.2.0, "COMMAND     EXT, myInput, I, 29
= -g -v halfgate_p halfgate_b     EOF
TIME = Tue Jul 16 21:53:31
1996"
PART, 4003PC84-4
SYM, XSYM1, OBUF, SLOW
    PIN, 0, 0, myOutput, 3.0
    PIN, I, I, _IN_myInput,
8.6, INV
END
SYM, XSYM2, IBUF
    PIN, 0, 0, _IN_myInput,
2.8
    PIN, I, I, myInput
END

```

8.3.2 Actel

The Actel files for the halfgate ASIC	
ADL file	STF file
<pre> ; HEADER ; FILEID ADL ./halfgate_io.adl 85e8053b ; CHECKSUM 85e8053b ; PROGRAM certify ; VERSION 23/1 ; ALSMAJORREV 2 ; ALSMINORREV 3 ; ALSPATCHREV .1 ; NODEID 72705192 ; VAR FAMILY 1400 ; ENDHEADER DEF halfgate_io; myInput, myOutput. USE ADLIB:INBUF; INBUF_2. USE ADLIB:OUTBUF; OUTBUF_3. USE ADLIB:INV; u2. NET DEF_NET_8; u2:A, INBUF_2:Y. NET DEF_NET_9; myInput, INBUF_2:PAD. NET DEF_NET_11; OUTBUF_3:D, u2:Y. NET DEF_NET_12; myOutput, OUTBUF_3:PAD. END. </pre>	<pre> ; HEADER ; FILEID STF ./halfgate_io.stf c96ef4d8 ... lines omitted ... (126 lines total) DEF halfgate_io. USE ; INBUF_2/U0; TPADH:'11:26:37', TPADL:'13:30:41', TPADE:'12:29:41', TPADD:'20:48:70', TYH:'8:20:27', TYL:'12:28:39'. PIN u2:A; RDEL:'13:31:42', FDEL:'11:26:37'. USE ; OUTBUF_3/U0; TPADH:'11:26:37', TPADL:'13:30:41', TPADE:'12:29:41', TPADD:'20:48:70', TYH:'8:20:27', TYL:'12:28:39'. PIN OUTBUF_3/U0:D; RDEL:'14:32:45', FDEL:'11:26:37'. END. </pre>

8.3.3 Altera

EDIF netlist in Altera format for the halfgate ASIC

```

(edif halfgate_p          (direction          (portRef
(edifVersion 2 0 0) OUTPUT)) myInput)
(edifLevel 0)            (designator          (portRef IN
(keywordMap              "@@Label"))))      (instanceRef
(keywordLevel 0))      (library working    B1_i1)))
(status                  (edifLevel 0)      (net myOutput
(written                 (technology        (joined
(timestamp 1996 7       (numberDefinition (portRef
10 23 55 8)            ) myOutput)
(program "COMPASS      (simulationInfo     (portRef OUT
Design Automation --   (logicValue H)      (instanceRef
EDIF Interface"       (logicValue         B1_i1))))
(version "v9r1.2 L)) (net VDD
last updated 26-Mar-   (cell halfgate_p   (joined )
96"))                 (cellType          (property
(author                GENERIC)          global
"mikes"))             (view              (string
(library flex8kd      COMPASS_nls_view   "vcc"))))
(edifLevel 0)         (viewType          (net VSS
(technology           NETLIST)            (joined )
(numberDefinition     (interface          (property
)                    (port myInput    global
(simulationInfo      (direction          (string
(logicValue H)      INPUT))            "gnd"))))))
(logicValue          (port myOutput    (design halfgate_p
L))                 (direction        (cellRef halfgate_p
(cell not            OUTPUT))          (libraryRef
(cellType           (designator        working))))
GENERIC)            "@@Label"))
(view               (contents
COMPASS_mde_view    (instance B1_i1
(viewType           (viewRef
NETLIST)           COMPASS_mde_view
(interface          (cellRef not
(port IN            (libraryRef
(direction         flex8kd))))
INPUT))            (net myInput
(port OUT          (joined

```

Report for the halfgate ASIC fitted to an Altera MAX 7000 complex PLD

** INPUTS **

Pin	LC	LAB	Primitive	Code	Shareable		Fan-In		Fan-Out		Name	
					Total	Shared	n/a	INP	FBK	OUT		FBK
43	-	-	INPUT		0	0	0	0	0	0	1	myInput

** OUTPUTS **

Pin	LC	LAB	Primitive	Code	Shareable		Fan-In		Fan-Out		Name	
					Total	Shared	n/a	INP	FBK	OUT		FBK
41	17	B	OUTPUT	t	0	0	0	1	0	0	0	myOutput

** LOGIC CELL INTERCONNECTIONS **

Logic Array Block 'B':

```

      +- LC17 myOutput
      |
LC    | | A B | Name

```

Pin

```

43   -> * | - * | myInput

```

* = The logic cell or pin is an input to the logic cell (or LAB) through the PIA.

- = The logic cell or pin is not an input to the logic cell (or LAB).

The structural postlayout files generated by the Altera MAX+plus software:

```

// halfgate_p (EPM7032LC44) MAX+plus II Version 5.1 RC6 10/03/94
// Wed Jul 17 04:07:10 1996
`timescale 100 ps / 100 ps

```

```

module TRI_halfgate_p( IN, OE, OUT );input IN; input OE; output OUT;
bufif1 ( OUT, IN, OE );
  specify
    specparam TTRI = 40; specparam TTXZ = 60; specparam TTZX = 60;
    (IN => OUT) = (TTRI,TTRI);
    (OE => OUT) = (0,0, TTXZ, TTZX, TTXZ, TTZX);
  endspecify
endmodule

```

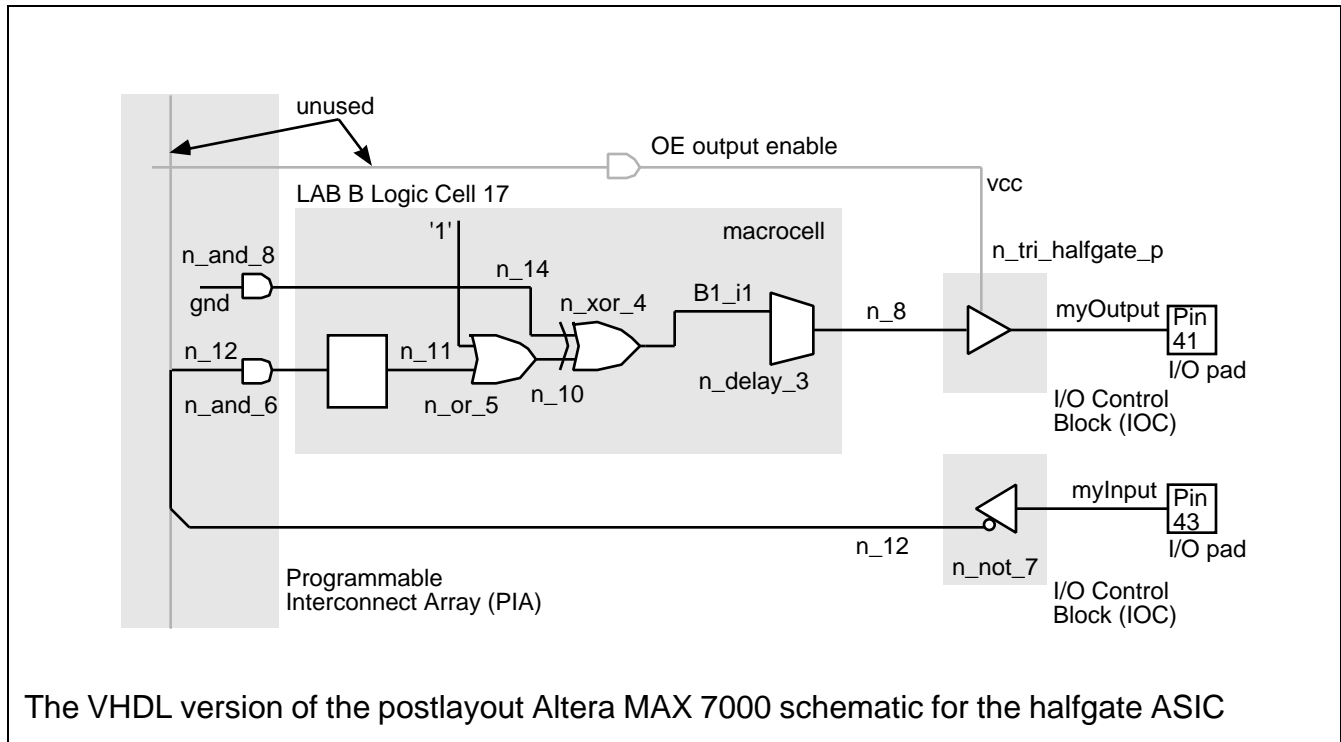
```

module halfgate_p (myInput, myOutput);
  input myInput; output myOutput; supply0 gnd; supply1 vcc;
  wire B1_i1, myInput, myOutput, N_8, N_10, N_11, N_12, N_14;
  TRI_halfgate_p tri_2 ( .OUT(myOutput), .IN(N_8), .OE(vcc) );
  TRANSPORT transport_3 ( N_8, N_8_A );
  defparam transport_3.DELAY = 10;
  and delay_3 ( N_8_A, B1_i1 );

```

```
xor xor2_4 ( B1_i1, N_10, N_14 );
or or1_5 ( N_10, N_11 );
TRANSPORT transport_6 ( N_11, N_11_A );
defparam transport_6.DELAY = 60;
and and1_6 ( N_11_A, N_12 );
TRANSPORT transport_7 ( N_12, N_12_A );
defparam transport_7.DELAY = 40;
not not_7 ( N_12_A, myInput );
TRANSPORT transport_8 ( N_14, N_14_A );
defparam transport_8.DELAY = 60;
and and1_8 ( N_14_A, gnd );
endmodule
```

```
// MAX+plus II Version 5.1 RC6 10/03/94 Wed Jul 17 04:07:10 1996
`timescale 100 ps / 100 ps
module TRANSPORT( OUT, IN ); input IN; output OUT; reg OUTR;
wire OUT = OUTR; parameter DELAY = 0;
`ifdef ZeroDelaySim
    always @IN OUTR <= IN;
`else
    always @IN OUTR <= #DELAY IN;
`endif
`ifdef Silos
    initial #0 OUTR = IN;
`endif
endmodule
```



8.3.4 Comparison

- Xilinx XC4000, a nondeterministic coarse-grained FPGA
- Actel ACT 3, a nondeterministic fine-grained FPGA
- Altera MAX 7000, a deterministic complex PLD

The differences:

1. The Xilinx LCA architecture does not permit an accurate timing analysis until after place and route. This is because of the coarse-grained nondeterministic architecture.
2. The Actel ACT architecture is nondeterministic, but the fine-grained structure allows fairly accurate preroute timing prediction.
3. The Altera MAX CPLD requires logic to be fitted to the product steering and programmable array logic. The Altera MAX 7000 has an almost deterministic architecture, which allows accurate preroute timing.

8.4 Summary

Key concepts:

- FPGA design flow: design entry, simulation, physical design, and programming
- Schematic entry, hardware design languages, logic synthesis
- PALASM as a common low-level hardware description
- EDIF, Verilog, and VHDL as vendor-independent netlist standards