ACT 1™ Family Gate Arrays
Product Guide

May 1989

Actel Corporation
955 East Arques Avenue
Sunnyvale, CA 94086-4520
(408) 739-1010

Data Sheets:

Application Notes:

Applications Briefs:

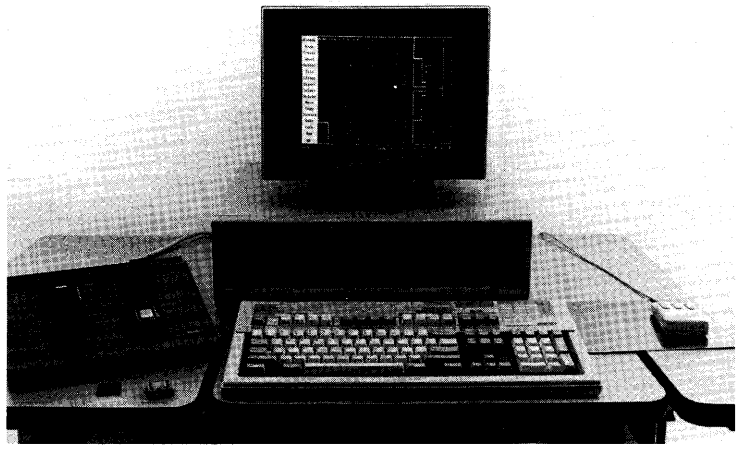
Data Sheets:

Action Logic System
ACT1 Family Gate Arrays
Actel Library

Action Logic™ System

Features:

- Extensive Gate Array Macro Library
- Supports Popular CAE systems
- Design Validation, Electrical Rules Check
- Fully Automated Place and Route
- Menu-driven Pin Editor for I/O Assignment
- Timing Analyzer Inspects all Critical Paths and AC Specifications



Designing with the Action Logic System

Overview

The Action Logic System is a high-productivity CAE environment for implementing logic functions with ACT devices. A menu-driven interface allows users to complete ACT™1 gate array designs, from concept to silicon, within hours without costly NRE. The system includes a software design environment and an Activator programmer, tester and debugger. Most popular CAE platforms are supported. Designers use their workstation to capture schematics, simulate, verify, place & route, perform timing analysis, program, and debug the chip at their desk. Actel's on-line help screens and reference manuals speed and simplify the design process.

Gate Array Macro Library

The Actel macro library contains the logic function building blocks necessary to create a design. The library includes macros ranging in complexity from simple gates to complex functions. Each macro has a graphic symbol or icon. Hard macros also contain placement information, a netlist and timing data.

Basic gates from the library are used to create soft macros like counters, adders and decoders. The Actel library contains over 150 different macros.

Schematic Capture and Simulation

Users enter their schematic using the Actel library. When completed, functional simulation is performed on the design.

Pin Editor

The Actel Pin Editor is an easy-to-use, menu-driven program for user assignment of logic I/Os and package pins. The editor automatically scrolls a list of all user-designated I/Os. To assign any pin to an I/O, the user enters the desired pin number next to the I/O node name. During each pin assignment, the Editor insures each pin is a valid package pin and is not already assigned to another chip I/O.

Actel Corporation, 955 East Arques Avenue, Sunnyvale, CA 94086-4520 408/7391010

Design Validation

The Actel Design Validator examines an ACT design for adherence to design rules specific to the ACT device chosen for the design. Validation verifies routability, and performs design rules checks prior to routing. The Validator produces error and information messages, and system warnings regarding electrical rules violations such as excessive fanout, shorted outputs, or unconnected inputs. A warning is issued, for example, if a net exceeds a fanout of ten; unconnected module inputs produce an error message.

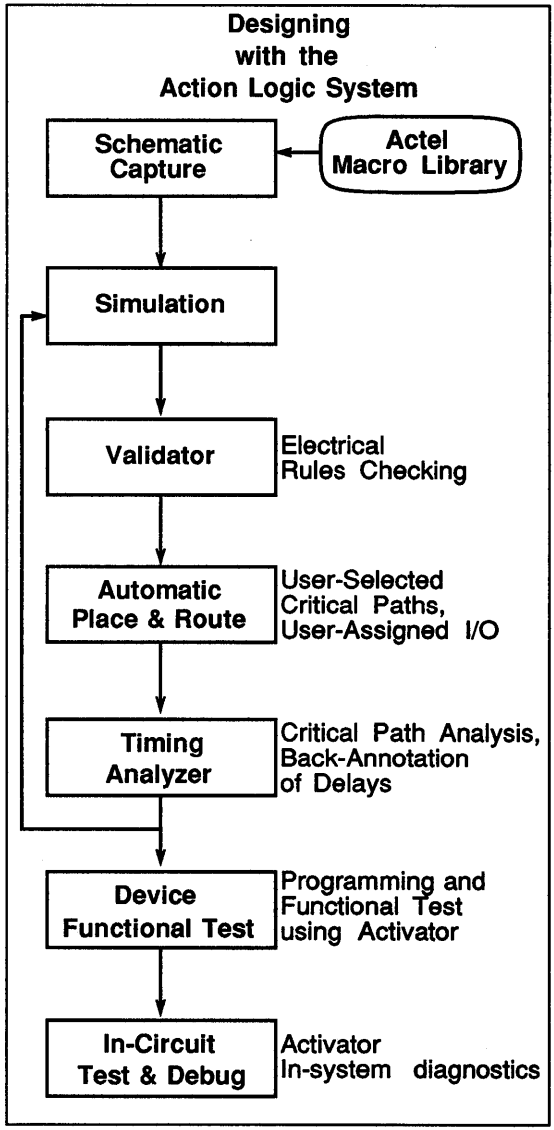
The Validator also provides statistical information about routability, logic module count, I/O count, average fanout per net, and array utilization. After passing design through the, the design is free of electrical rules violations.

Automatic Place & Route

Fully automatic place and route software minimizes design delay by assigning macros to optimal locations in the chip. The system uses the design's netlist, critical net information and I/O assignments to automatically place and route all the logic blocks within the circuit.

The software provides the user with data on actual wire length, capacitive loading and wiring congestion. The route program assigns the shortest possible net segments to connect the library macros with minimal delay, routing 100% of the nets automatically for 85-95% logic module utilization. The program provides data on actual wire length, capacitive loading and wiring congestion. No manual intervention is required, even at high device utilization.

Actel Place and Route takes advantage of The ACT device's efficient architecture and abundance of routing resources to consistently place and route 100% of the nets.



Timing Analysis

The timing analyzer is an interactive tool which determines and highlights all critical and non-critical paths within a specified design. After the layout phase is complete, the Timing Analyzer extracts accurate net delays. These delays are back-annotated to the netlist and evaluated with the timing analyzer or simulator. It provides data concerning the slowest paths in the design and any other user specified paths. Using this information, the designer optimizes the design to meet its timing specifications. The Timer accepts as its input the design's netlist and delay information. The user directs the type of analysis and output. Delay reports generated by the Timing Analyzer using post-route numbers provide the final AC specifications for the designs.

Device Programming

Programming is controlled by the Activator™ programming system. Action Logic software generates a fuse map for the ACT device which is used by the Activator for programming. A series of internal address registers are automatically loaded specifying the programming element (PLICE antifuse) to be programmed. A programming sequence is then initiated, creating a permanent link. These steps are repeated until all interconnections are made.

Functional Test

The Activator supports functional testing using an I/O test vector file. It also supports interactive circuit debugging.

In-circuit Test and Debug

Once the device is programmed and functionally verified, it is ready for operation in the user's system. The Actionprobe diagnostics may then be used to further evaluate circuit integrity.

ACT™1 Family
Desktop Configurable Gate Arrays

Features:

- **High Gate and I/O Count:**
  ACT1010-- 1200 gates, 57 I/Os
  ACT1020-- 2000 gates, 69 I/Os
- **Instant Prototypes and Production**
- **Toggle rates to 70 MHz**
- **System-Level Performance to 40 MHz**
- **Gate Array Architecture Allows Completely Automatic Place and Route**
- **Non-Volatile, Permanent Programming**
- **Built-In Clock Distribution Network**
- **Built-In Diagnostic Probes**
- **Low Power CMOS technology**
- **Fully supported by Actel's Action Logic™ System**

ACT1010/1020 Description

ACT1010 and ACT1020 devices are the first members of a family of Application Configurable Technology (ACT) gate arrays offered by Actel. These devices are implemented in silicon gate, 2 micron, 2-level metal CMOS, employing Actel's PLICE™ anti-fuse technology. The unique architecture offers gate array flexibility, high performance and instant turnaround through user programming. Utilizations as high as 90% permit designs of 1200 equivalent gates for the ACT1010, and 2000 gates for the ACT1020.

The ACT1010/1020 also provide system designers with unique on-chip diagnostic probe capabilities allowing convenient testing and debugging. Additional features include an on-chip clock driver with a hard wired distribution network. The network provides efficient clock distribution with minimum skew.

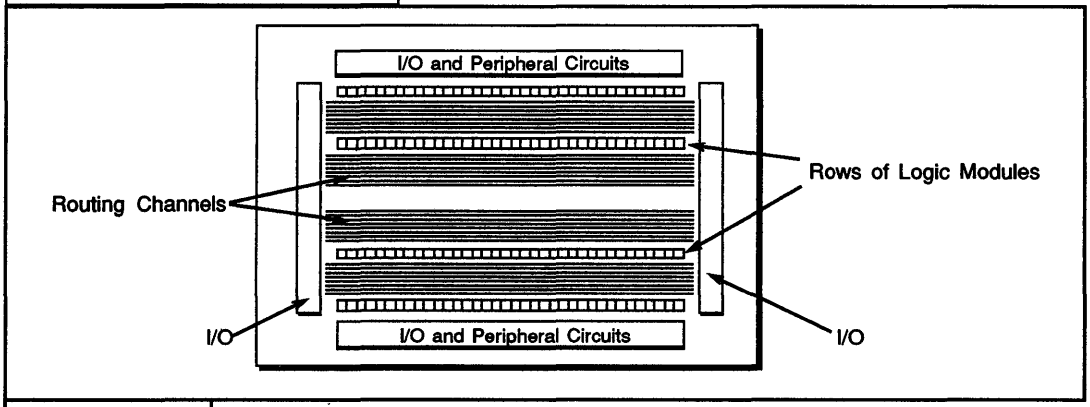
The user-configurable I/Os are capable of driving at both TTL and CMOS drive levels. The ACT1010/1020 are available in 44, 68 and 84 pin leaded chip carriers, and 84PGA.

The ACT device signature register offers programmable identification for both part type and user ID information. A security fuse may be used to permanently disable all further programming, diagnostics and testing once the chip is configured. Designs are thus protected from copying or reverse engineering.

Action Logic System Description

ACT1010/1020 devices are supported by Actel's Action Logic System, providing logic design implementation with minimum effort. The Action Logic System interfaces with the resident CAE system to provide a complete gate array design environment: schematic capture, simulation, fully automatic place and route, timing verification and device programming. The Action Logic System is available for 386 PC and compatible workstations, including Apollo and Sun workstations.

Figure 1: ACT Device Architecture



ACT 1010/1020

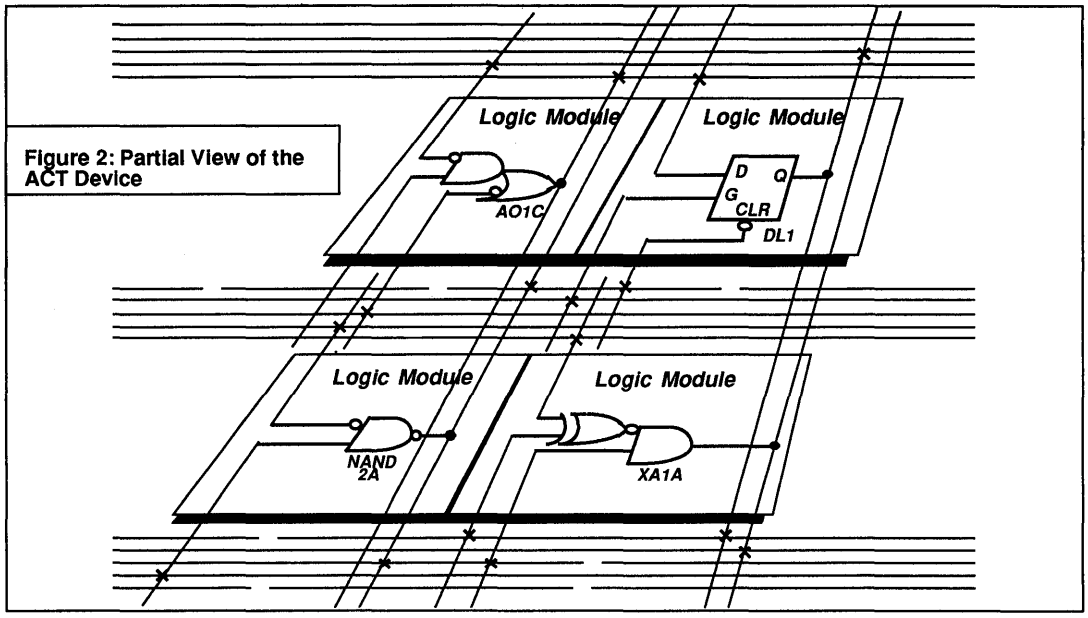


Figure 2: Partial View of the ACT Device

ACT Device Structure

A partial view of an ACT device (Figure 2) shows four logic modules and distributed horizontal and vertical interconnect tracks. Logic module inputs, outputs, and track segments are connected via PLICE antifuses located at intersections. During programming, these antifuses are addressed and programmed to make the connections required by the circuit application.

The Actel Logic Module

The Actel logic module is an eight input, one output configurable logic circuit chosen for the wide range of functions it implements and for its efficient use of interconnect routing resources.

The logic module can implement the four basic logic functions (NAND, AND, OR, and NOR) in gates of two, three or four inputs. Each function may have many versions, with different combinations of active-low inputs. The logic module can also implement a variety of D-latches, exclusivity function, AND-ORs, and OR-ANDs. No dedicated hardwired latches or flip-flops are required in the array since latches and flip-flops may be constructed from logic modules wherever needed in the application.

I/O Buffers

Each I/O pin is configurable as an input, output, three-state, or bi-directional buffer. Input and output levels are compatible with standard TTL and CMOS specifications. Outputs sink or source 4 mA at TTL levels. See the DC Characteristics for additional I/O buffer specifications.

Device Organization

ACT devices consist of a matrix of logic modules arranged in rows separated by wiring channels (Figure 1). This array is surrounded by a ring of peripheral circuits including I/O buffers, testability circuits, and diagnostic probe circuits providing real-time diagnostic capability. Between rows of logic modules are routing channels containing sets of segmented metal tracks with PLICE antifuses. Each channel has 22 signal tracks. The resulting network allows arbitrary and flexible interconnections between logic modules and to I/O modules.

Actionprobe Diagnostics

ACT1010/1020 devices each have two independent diagnostic probe pins. These pins allow the user to observe any two internal signals by entering the appropriate net name in the diagnostic software. Signals may be viewed on an oscilloscope, logic analyzer or with the Action Logic System. The probe pins can also be used as user-defined I/Os, according to the level of the mode control pin. When configured as user-defined I/Os, the pins have the same characteristics as other I/O pins.

ACT Array Performance

Worst-Case Delay Conversion

Worst-case delays for ACT arrays are calculated in the same manner as for masked array products. Typical delay parameters are multiplied by factors to account for temperature and voltage effects. However, in an ACT array, temperature and voltage effects are less dramatic than with masked devices. This is due to the fact that the electrical characteristics of module interconnections on ACT devices remain constant over voltage and temperature fluctuations. Therefore, unlike masked devices, performance variations of ACT arrays from voltage and temperature changes are due only to changes in the active devices.

As a result, the total derating factor from typical to worst case for an ACT array is only 1.18 to 1, compared to 2 to 1 for a masked gate array.

Logic Module Size

Logic module size also affects performance. A mask programmed gate array cell with four transistors usually implements only one logic level. In an ACT array's more complex logic module (similar to the complexity of a gate array macro), implementation of multiple logic levels within a single module is possible. This eliminates interlevel wiring and associated RC delays. The effect is termed "net compression."

Derating Curves

Logic module and storage element delays are shown in the tables on page 5. These delays may be used directly from the table for estimating purposes. Temperature and voltage are derated according to the curves in the graphs on page 7. The Action Logic System timing analyzer provides actual timing specifications for each circuit implementation.

Resources

	ACT1010	ACT1020
Logic Modules	295	546
User Defined I/O	57*	69*

*Includes diagnostic pins

Operating Characteristics

Absolute Maximum Ratings*

Parameter	Symbol	Limits	Units
DC Supply Voltage	VCC	-0.3 to +7	Volts
Input Level	V_I	-0.3 to VCC +0.3	Volts
Storage Temp Range	T_{STG}	-40 to +125	°C

* Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

Recommended Operating Conditions

Parameter	Symbol	Limits	Units
DC Supply Voltage	VCC	4.75 to 5.25	Volts
Program Voltage	VPP	VCC*	Volts
Operating Ambient	T_A	0 to 70	°C

*Except during programming

DC Operating Characteristics

VCC= 5V±5%, TA=0°C to 70°C

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IH}	HIGH level input	TTL inputs	2.00		V
V_{IL}	LOW level input	TTL inputs	0.0	0.8	V
V_{OH}	HIGH level output (Note 1)	I_{OH}= -4mA DC	3.7		V
V_{OL}	LOW level output (Note 1)	I_{OL}= 4mA DC		0.45	V
I_{IN}	Input leakage	V_I=VCC or GND	-10	10	µA
I_{OZ}	3-state output leakage	V_O=VCC or GND	-10	10	µA
I_{DD1}	VCC supply (standby)	V_I=VCC or GND, I_o= 0mA		10	mA
I_{OS}	Output short circuit	V_O = VCC	20	140	mA
		V_O = 0V	-10	-100	mA

Capacitance (Notes 2, 3)

Symbol	Parameter	Conditions	Typ	Unit
C_{IO}	I/O Capacitance	V_O=0v; f=1.0MHz	10	pF

Notes:
1. Outputs tested one at a time.
2. Sample tested
3. Includes worst case 84LCC package pin capacitance.

AC Characteristics — Commercial Temperature & Voltage (VCC= 5V ± 5% TA 0° + 70°)

V_{IL}=OV; V_{IH}=3; V; V_{OL} = V_{OH}= 1.5V

Macro Description	Fanout	1		2		3		4		8	
		MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX
Gates Using 1 Module	t_{PD_1}	6.1	9.3	6.8	9.9	7.6	11.4	8.4	12.8	11.2	16.0
Gates Using 2 Modules	t_{PD_2}	10.1	16.0	10.8	16.6	11.6	18.1	12.4	19.5	15.2	22.7
Latches	t_{PD}	6.1	9.3	6.8	9.9	7.6	11.4	8.4	12.8	11.2	16.0
	t_{SU}	4.1	5	4.5	5.5	4.9	6	5.3	6.5	5.7	7
	t_H	4.1	5	4.9	6	5.7	7	6.5	8	7.3	9
Flip-flops Note: T_{SU} is independent of fanout.	t_{PD}	6.1	9.3	6.8	9.9	7.6	11.4	8.4	12.8	11.2	16.0
	t_{SU}	4.1	5	4.1	5	4.1	5	4.1	5	4.1	5
	t_H	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Input Buffer	t_{PLH}	7.8	8.9	8.1	9.6	9.4	11.2	10.2	12.1	14.6	17.3
	t_{PHL}	8.6	10.2	9.4	11.1	10.8	12.8	11.9	14.1	16.7	19.8

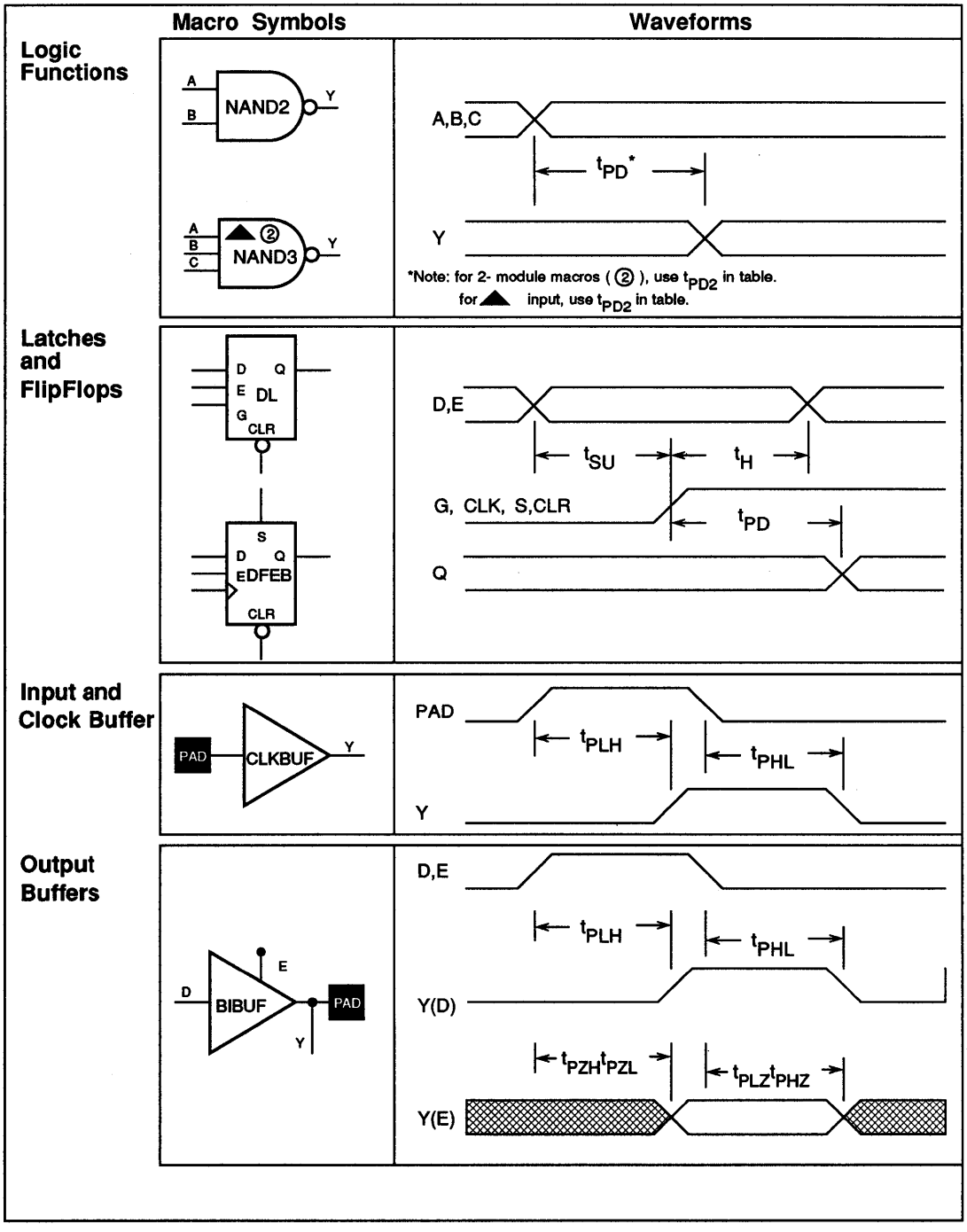
		MIN	MAX
Clock Buffer	t_{LH}	11	15
	t_{HL}	12	16

		50 PF		NS/PF	
		MIN	MAX	MIN	MAX
Output Buffers	t_{PLH}	5.6	6.9	0.04	0.06
	t_{PHL}	7.0	8.6	0.04	0.06

			MIN	MAX	MIN	MAX
Output Buffers, TS and Bidirectional	D	t_{PLH}	5.6	6.9	0.04	0.06
		t_{PHL}	7.0	8.6	0.04	0.06
	E	t_{PHZ}^{PZH}	5.2	6.9		
		t_{PLZ}^{PZL}	7.0	8.6		

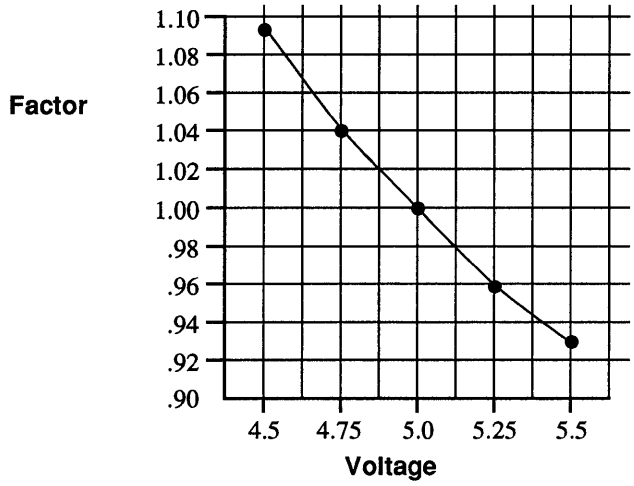
Note:

1. All T_{PD} include interconnect delays.
2. All delays include statistical estimates for wiring delay; actual delay values are determined after place and route.


Macro Symbols
Waveforms
Logic Functions
A
B
NAND2
Y
A,B,C
tPD*
A
B
C
②
NAND3
Y
Y
*Note: for 2- module macros (②), use tPD2 in table.
for ▲ input, use tPD2 in table.
Latches and FlipFlops
D
E
G
DL
Q
CLR
D,E
tSU
tH
S
D
EDFEB
Q
CLR
G, CLK, S,CLR
tPD
Q
Input and Clock Buffer
PAD
CLKBUF
Y
PAD
tPLH
tPHL
Y
Output Buffers
D,E
E
D
BIBUF
PAD
Y
tPLH
tPHL
Y(D)
tPZH tPZL
tPLZ tPHZ
Y(E)

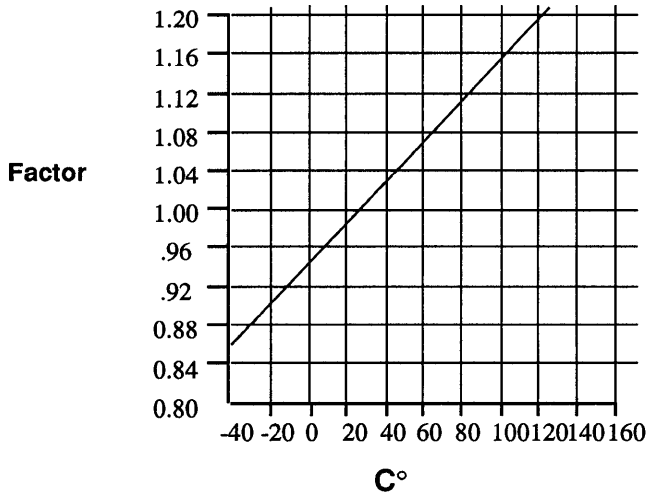
Derating Curves

Voltage



Voltage	Factor
4.50	1.095
4.75	1.046
5.25	0.960
5.50	0.930

Temperature



Determining ACT1
Dyanmic Power Calculations

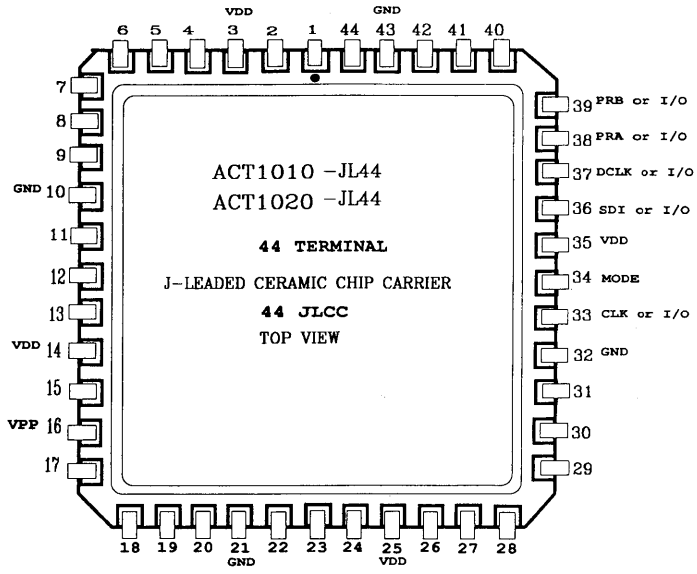
The following is the formula for calculating typical dynamic chip power consumption in milliwatts.

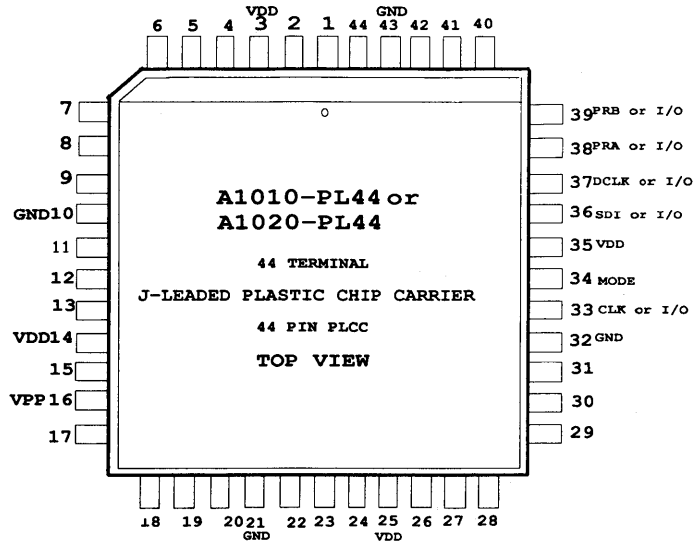
Total Chip Power = 0.15N*F1 + 0.012M*F2 + 2.6P*F3

Where:
F1 = Average logic module switching rate in MHz
F2 = Average clock pin switching rate in MHz
F3 = Average I/O switching rate in MHz
M = No. of logic modules connected to clock pin
N = Number of logic modules used on the chip
P = Number of I/O pairs used (input + output), with 50 pF load.

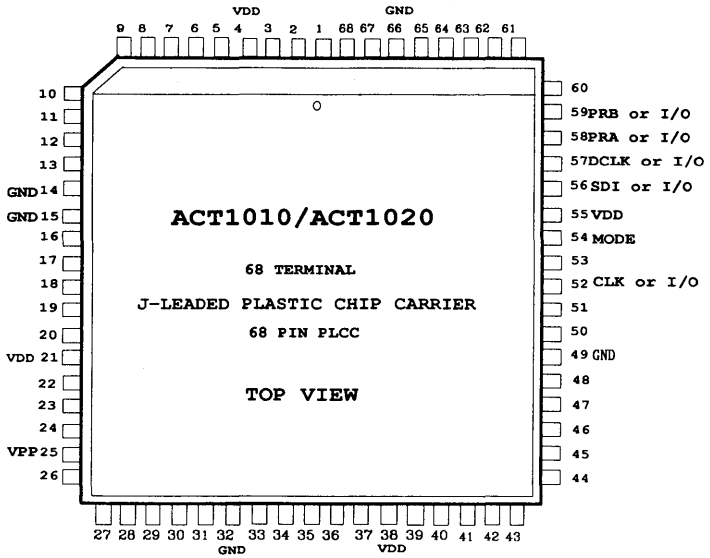
Note:
The power is calculated for a typical interconnect and a fanout of 3 from each of the logic modules.

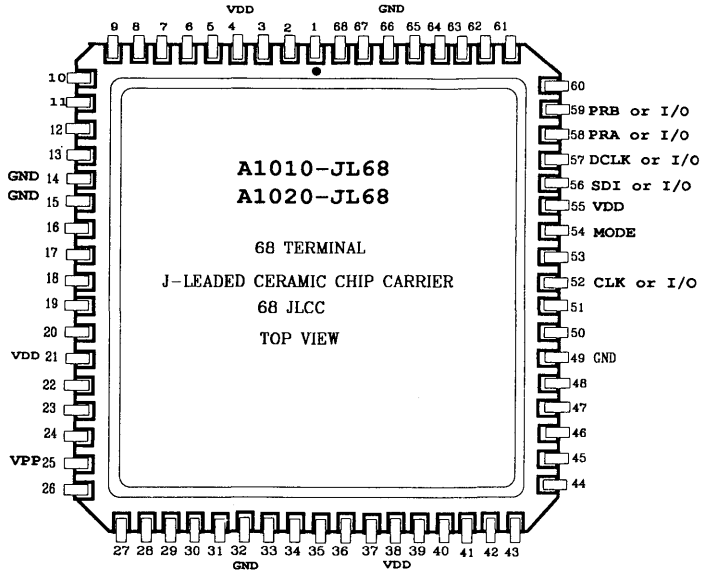
Packages


VDD
GND
6 5 4 3 2 1 44 43 42 41 40
7
8
9
GND 10
11
12
13
VDD 14
15
VPP 16
17
18 19 20 21 22 23 24 25 26 27 28
GND
VDD
39 PRB or I/O
38 PRA or I/O
37 DCLK or I/O
36 SDI or I/O
35 VDD
34 MODE
33 CLK or I/O
32 GND
31
30
29
ACT1010 –JL44
ACT1020 –JL44
44 TERMINAL
J–LEADED CERAMIC CHIP CARRIER
44 JLCC
TOP VIEW

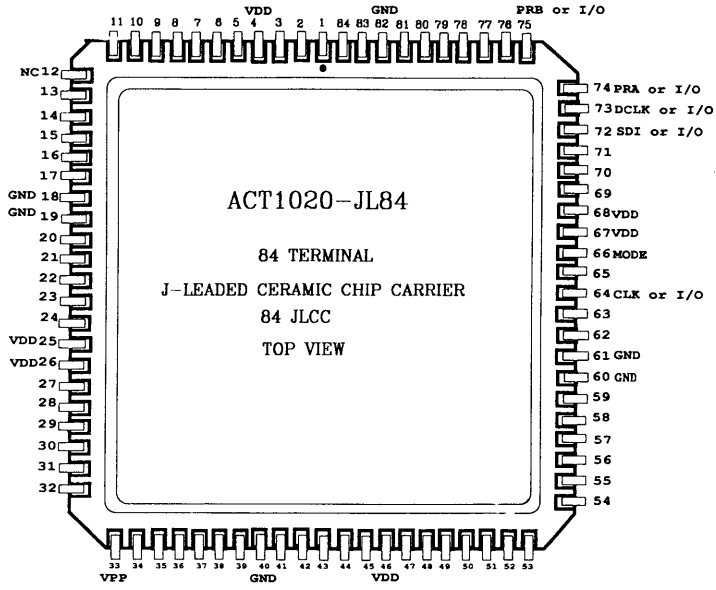

VDD
GND
6 5 4 3 2 1 44 43 42 41 40
7
8
9
GND10
11
12
13
VDD14
15
VPP16
17
18 19 20 21 22 23 24 25 26 27 28
GND
VDD
39 PRB or I/O
38 PRA or I/O
37 DCLK or I/O
36 SDI or I/O
35 VDD
34 MODE
33 CLK or I/O
32 GND
31
30
29
A1010–PL44 or
A1020–PL44
44 TERMINAL
J–LEADED PLASTIC CHIP CARRIER
44 PIN PLCC
TOP VIEW

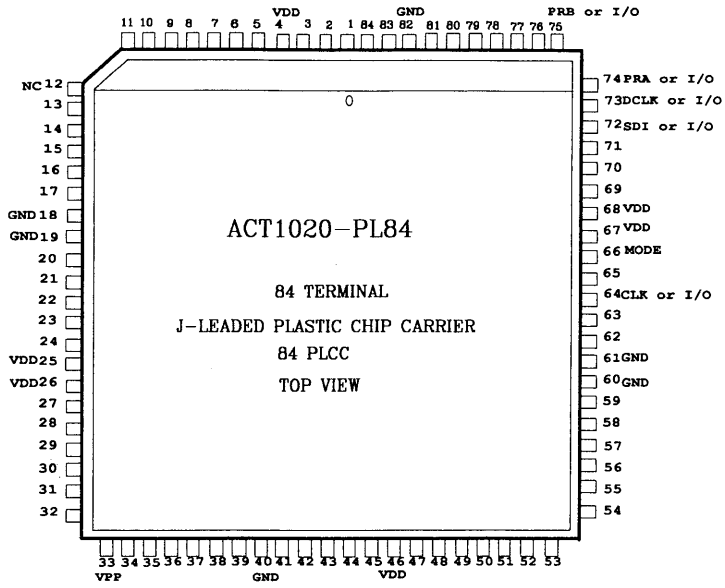
Packages


VDD
GND
9 8 7 6 5 4 3 2 1 68 67 66 65 64 63 62 61
10
11
12
13
GND 14
GND 15
16
17
18
19
20
VDD 21
22
23
24
VPP 25
26
60
59 PRB or I/O
58 PRA or I/O
57 DCLK or I/O
56 SDI or I/O
55 VDD
54 MODE
53
52 CLK or I/O
51
50
49 GND
48
47
46
45
44
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
GND
VDD
ACT1010/ACT1020
68 TERMINAL
J-LEADED PLASTIC CHIP CARRIER
68 PIN PLCC
TOP VIEW

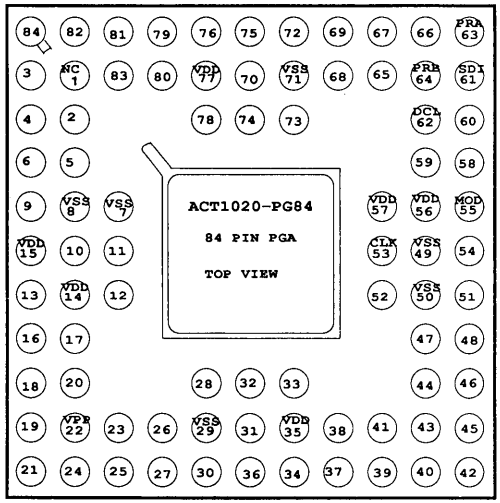

VDD
GND
9 8 7 6 5 4 3 2 1 68 67 66 65 64 63 62 61
10
11
12
13
GND 14
GND 15
16
17
18
19
20
VDD 21
22
23
24
VPP 25
26
60
59 PRB or I/O
58 PRA or I/O
57 DCLK or I/O
56 SDI or I/O
55 VDD
54 MODE
53
52 CLK or I/O
51
50
49 GND
48
47
46
45
44
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
GND
VDD
A1010-JL68
A1020-JL68
68 TERMINAL
J-LEADED CERAMIC CHIP CARRIER
68 JLCC
TOP VIEW

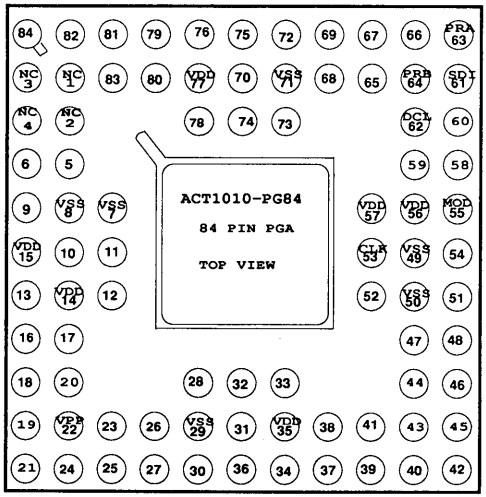
Packages


VDD
GND
PRB or I/O
11 10 9 8 7 6 5 4 3 2 1 84 83 82 81 80 79 78 77 76 75
NC 12
13
14
15
16
17
GND 18
GND 19
20
21
22
23
24
VDD 25
VDD 26
27
28
29
30
31
32
74 PRA or I/O
73 DCLK or I/O
72 SDI or I/O
71
70
69
68 VDD
67 VDD
66 MODE
65
64 CLK or I/O
63
62
61 GND
60 GND
59
58
57
56
55
54
33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
VPP
GND
VDD
ACT1020–JL84
84 TERMINAL
J–LEADED CERAMIC CHIP CARRIER
84 JLCC
TOP VIEW

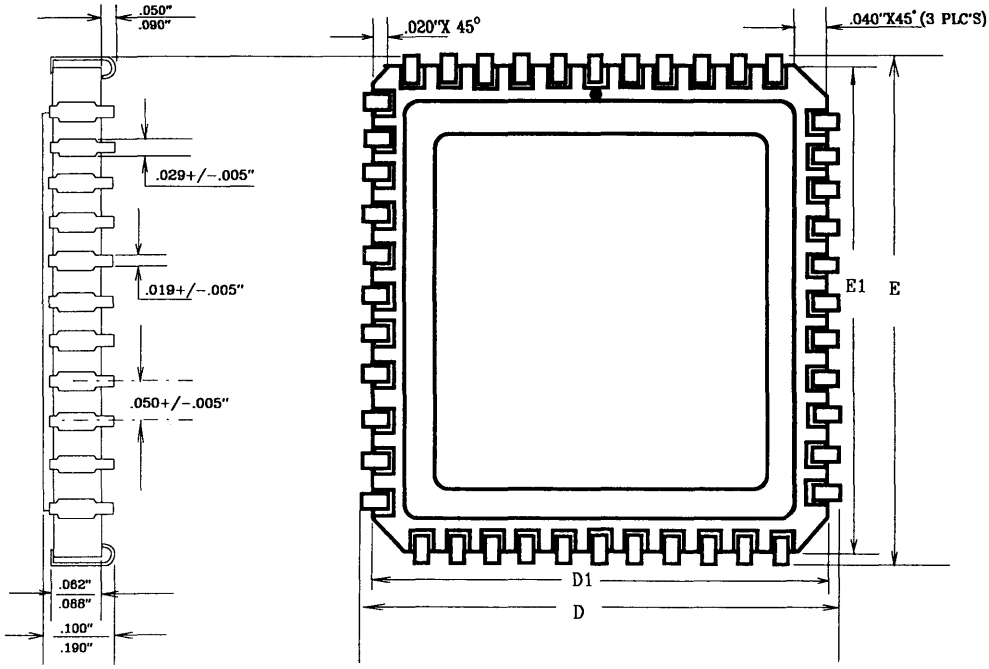

VDD
GND
PRB or I/O
11 10 9 8 7 6 5 4 3 2 1 84 83 82 81 80 79 78 77 76 75
NC 12
13
14
15
16
17
GND 18
GND 19
20
21
22
23
24
VDD 25
VDD 26
27
28
29
30
31
32
74 PRA or I/O
73 DCLK or I/O
72 SDI or I/O
71
70
69
68 VDD
67 VDD
66 MODE
65
64 CLK or I/O
63
62
61 GND
60 GND
59
58
57
56
55
54
33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53
VPP
GND
VDD
ACT1020–PL84
84 TERMINAL
J–LEADED PLASTIC CHIP CARRIER
84 PLCC
TOP VIEW

Packages


ACT1020-PG84
84 PIN PGA
TOP VIEW


ACT1010-PG84
84 PIN PGA
TOP VIEW

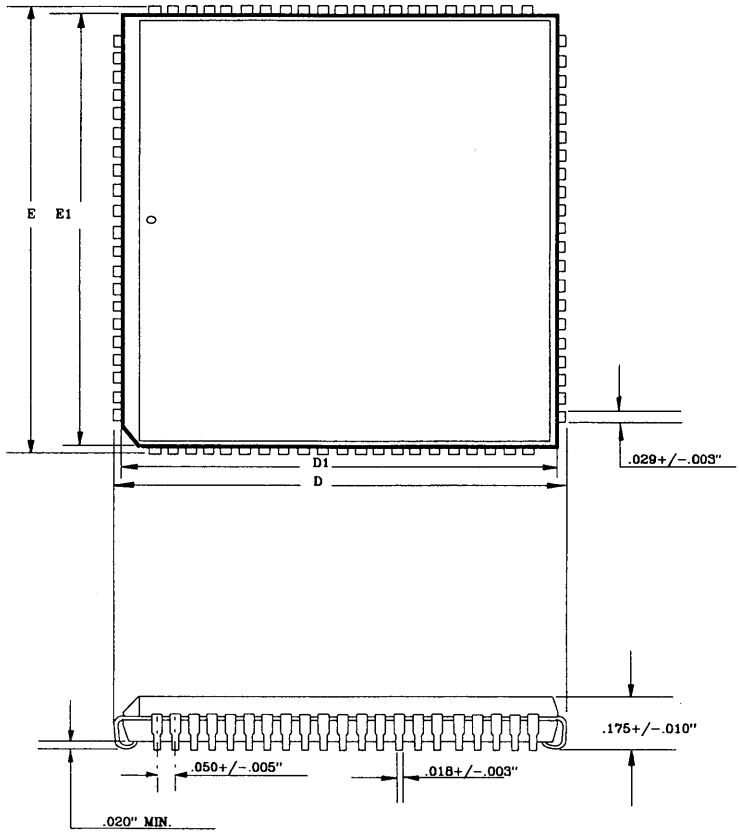
Package Dimensions



J-LEADED CHIP CARRIERS (JLCC)

LEAD COUNT	FORM WIDTH (D, E)		BODY WIDTH (D1, E1)	
	NOMINAL	TOLERANCE	NOMINAL	TOLERANCE
44	.690"	.010"	.650"	.010"
68	.990"	.010"	.950"	.010"
84	1.190"	.010"	1.150"	.010"

Package Dimensions

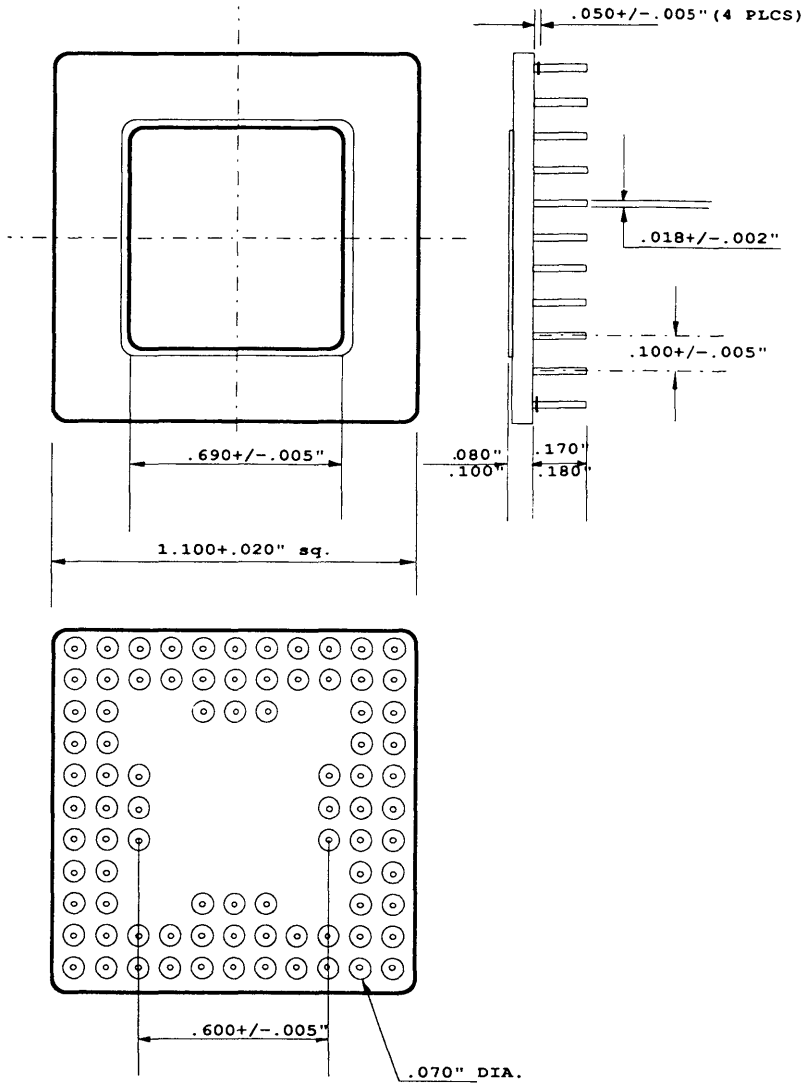


PLCC PACKAGE

LEAD COUNT	D, E	E1, D1
44	690+/-.005"	.655+/-.005"
68	.990+/-.005"	.955+/-.005"
84	1.190+/-.005"	1.155+/-.005"

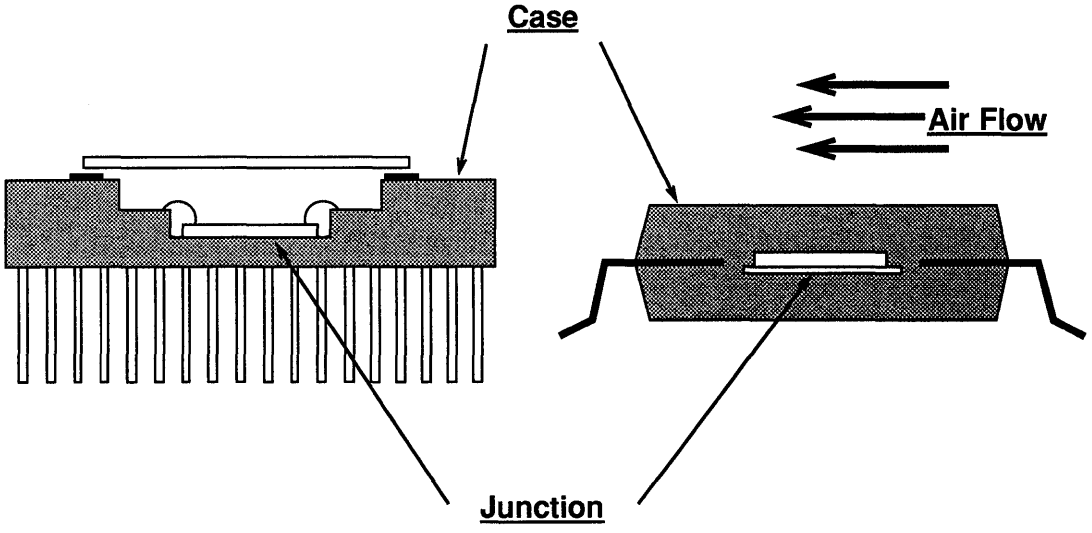
Package Dimensions

84 PIN PGA OUTLINE DRAWING



ACT 12XXX Package Thermal Resistance

Package Type	Pin Count	Øjc	Øja still air	Øja 300 ft/min
Ceramic PGA	84	8	33	20
Plastic LCC	44	15	52	40
	68	13	45	35
	84	12	44	33
Ceramic JLCC	44	5	38	30
	68	5	38	25
	84	7	34	24



$\emptyset jc = \emptyset ja = \Delta T \emptyset ja$

Notes on package pin designations:

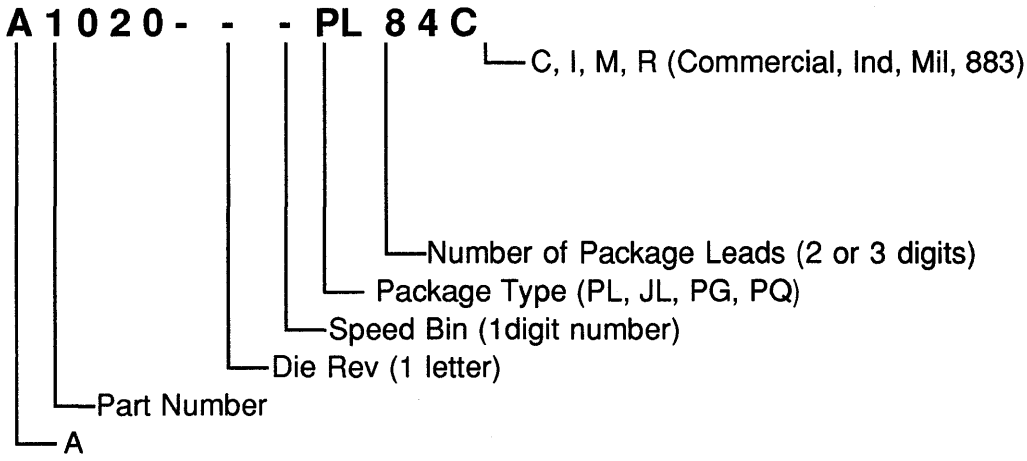
1. All pins marked GND are I/O ground connections and must be connected to circuit ground.
2.. All pins marked GNDA are internal array grounds and must be connected to circuit ground.
3. V_{PP} must be terminated to VCC except during programming.
4. PRA and PRB, the diagnostic probe outputs, should remain open if not used as I/Os.
5. MODE must be terminated to circuit ground except during programming*.
6. SDI and DCLK should be terminated to circuit ground during normal operation if not used as I/Os*.
7. Unused I/O pins are automatically designated by the Action Logic System as outputs and should remain unconnected. They will be driven low by the device.
8. All unidentified pins on the package outline drawings are standard I/Os.

* Note:

The security fuse must be programmed for SDI and DCLK to function as I/Os.

For device debugging on the user's circuit board, MODE, SDI and DCLK should be terminated to circuit ground through a ≥ 10K ohm resistor. They can be tied to ground if not debugging.

Ordering Information:



Unused parts are left out when identifying a product.

Action Logic System
Macro Library

Actel Macro Library

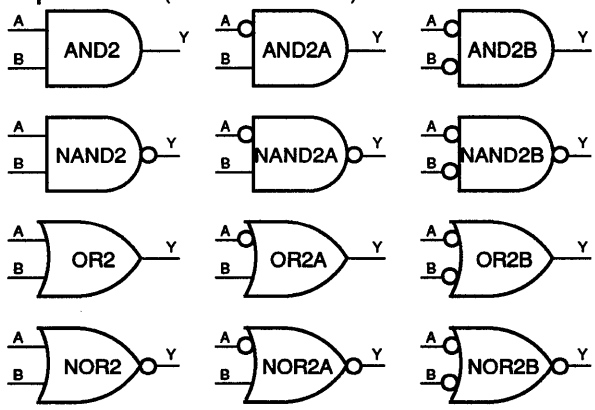
The ACT1010/1020 is supported by a macro library of over 150 standard logic functions. These range from simple logic gates to complex functions such as counters, decoders and comparators. All Library components are hard macros with fixed characteristics uness otherwise specified.

The Actel logic module implements logic functions with inverted inputs as efficiently as non-inverted inputs, without an increase in propagation delay. By taking advantage of the various combinations of input polarity, the use of separate inverters can be virtually eliminated.

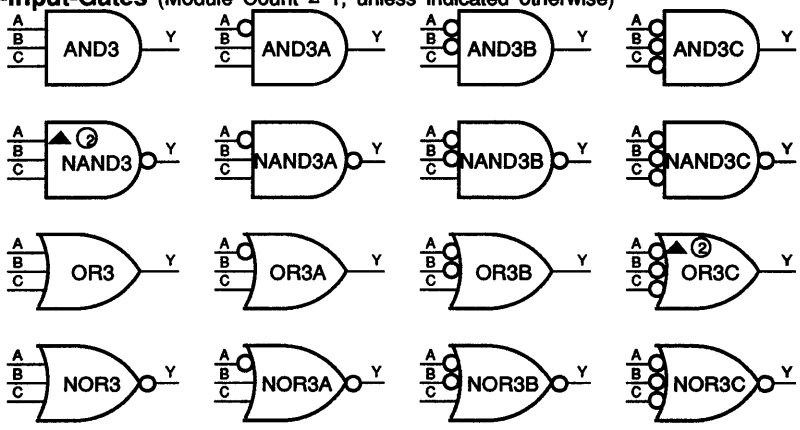
Actel Hard Macros

② Indicates 2-module macro
▲ Indicates extra delay input

2-Input-Gates (Module Count = 1)



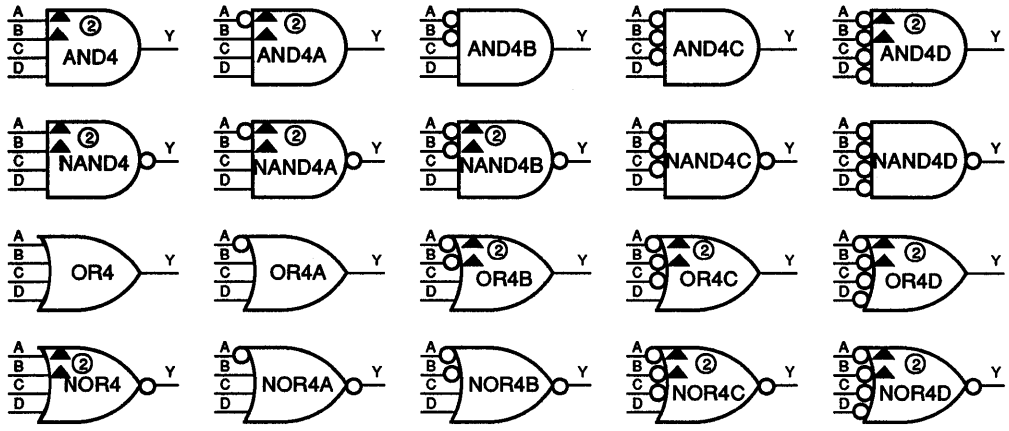
3-Input-Gates (Module Count = 1, unless indicated otherwise)



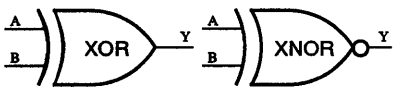
Actel Hard Macros

② Indicates 2-module macro
▲ Indicates extra delay input

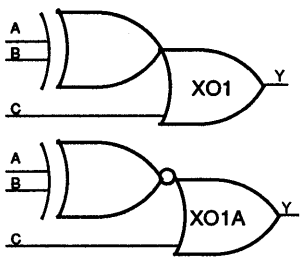
4-Input-Gates (Module Count = 1, unless indicated otherwise)



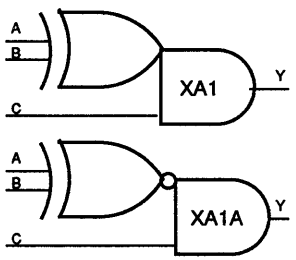
XOR Gates
(Module Count = 1)



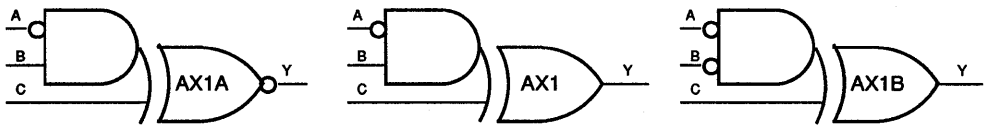
XOR OR Gates
(Module Count = 1)



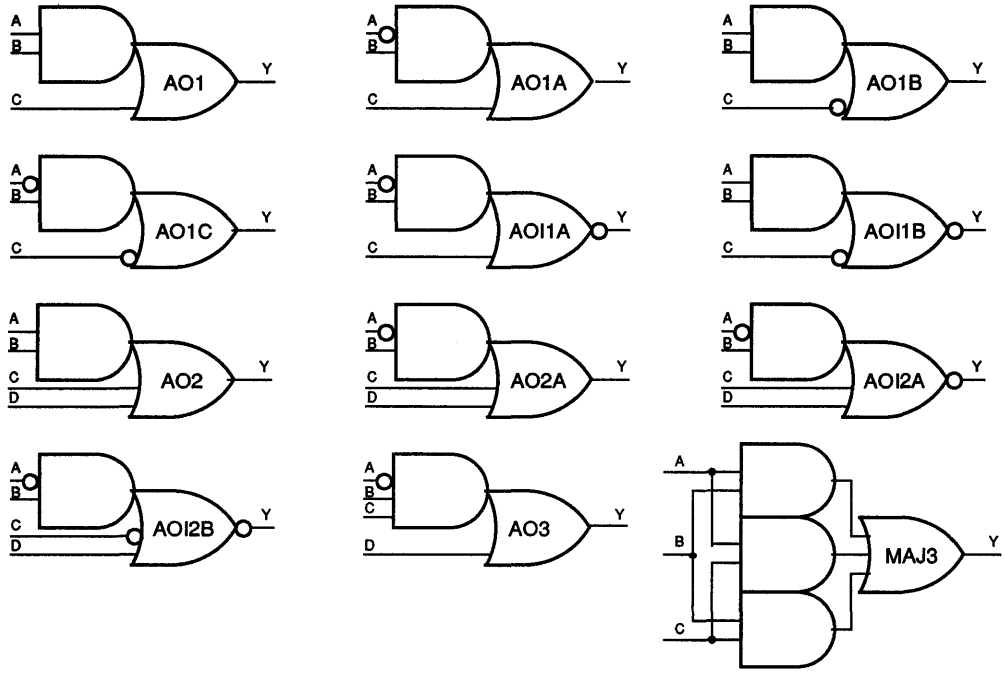
XOR AND Gates
(Module Count = 1)



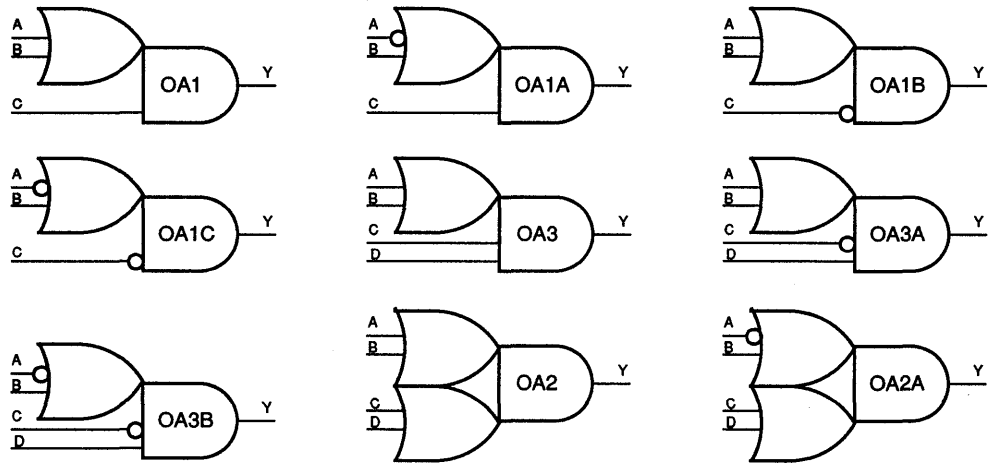
AND XOR Gates
(Module Count = 1)



AND OR Gates (Module Count = 1)

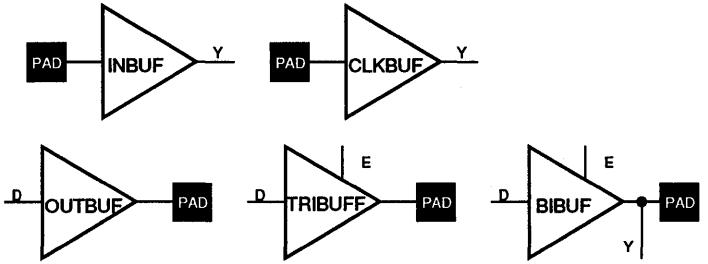


OR AND Gates (Module Count = 1)

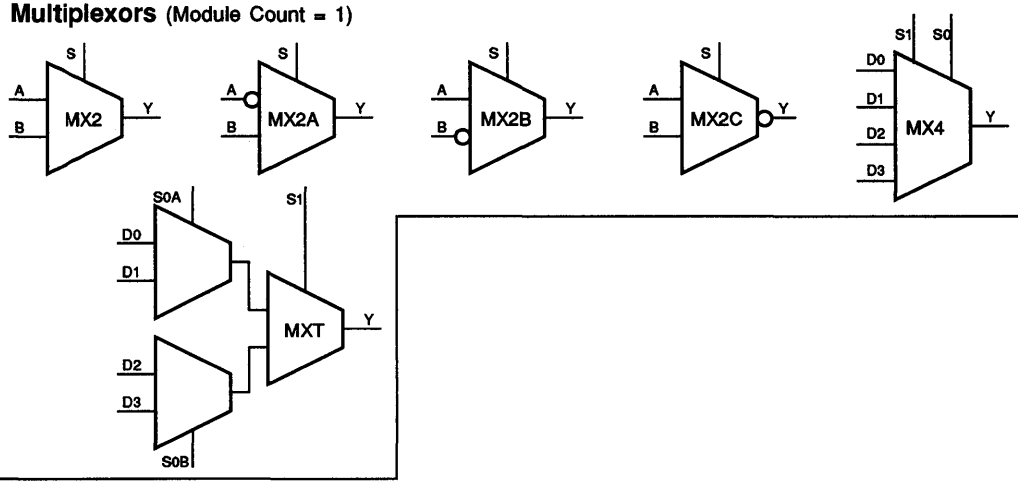


Buffers (Module Count = 1)

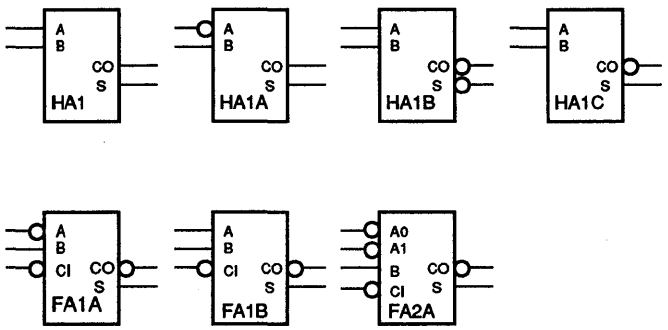
I/O Buffers (I/O MODULE Count = 1)



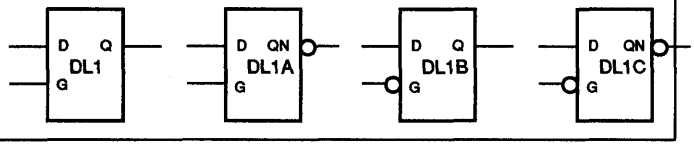
Multiplexors (Module Count = 1)



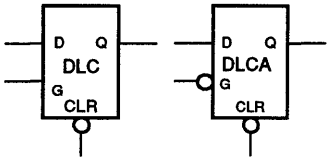
Adders (Module Count = 2)



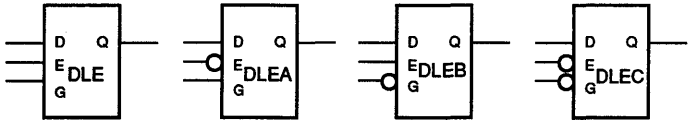
D Latches (Module Count = 1)



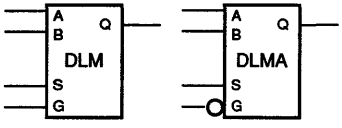
D Latches w/ Clear (Module Count = 1)



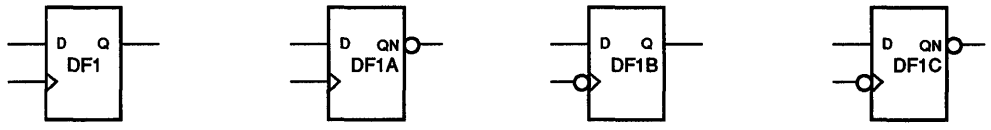
D Latches w/ Enable (Module Count = 1)



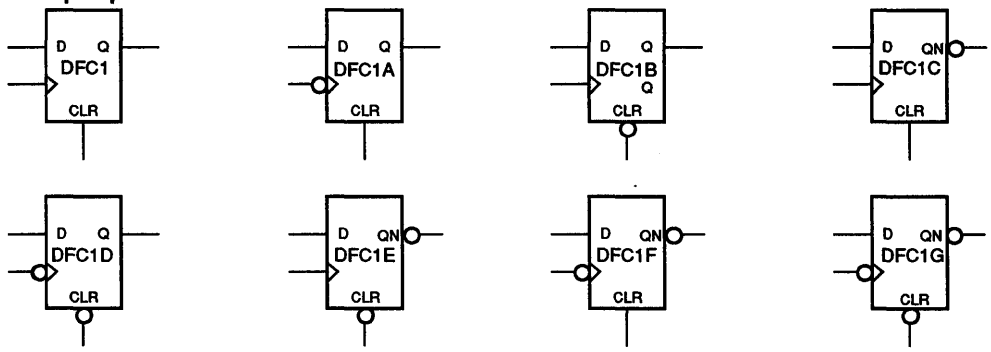
Mux Latches
(Module Count = 1)



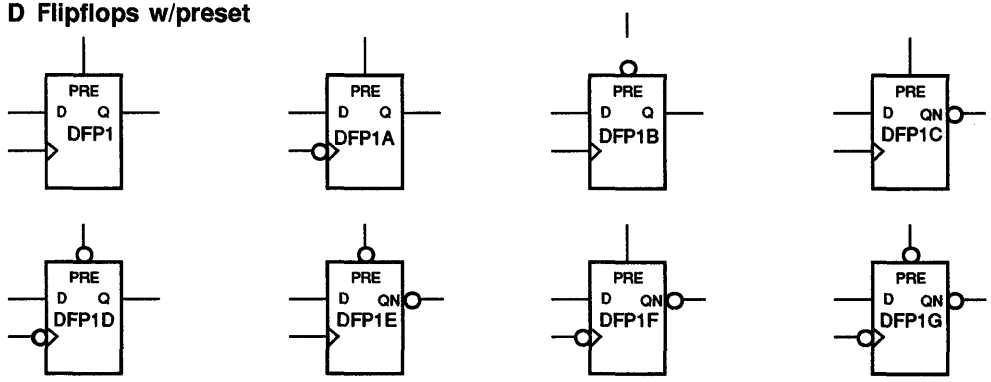
D Flipflops (Module Count = 2)



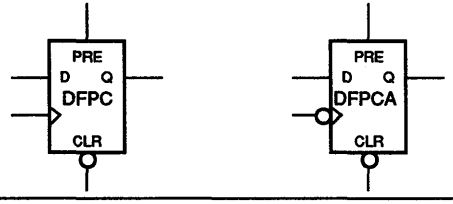
D Flipflops w/clear



D Flipflops w/preset

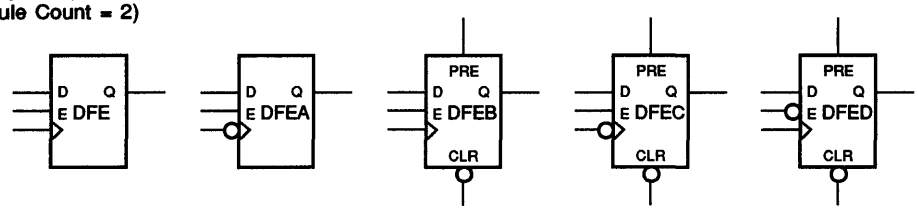


D Flipflops w/preset and clear



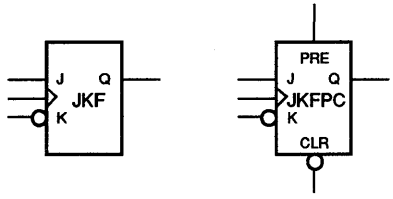
D Flip-Flops with Enable

(Module Count = 2)



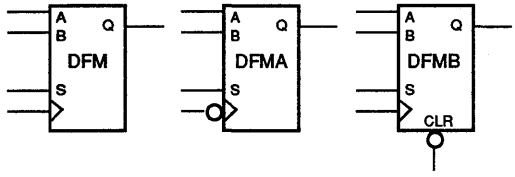
JK Flip-Flops

(Module Count = 2)



Mux Flip-Flops

(Module Count = 2)



Actel Soft Macros

Macro Name	Description	Module Count
TTL Equivalents:		
T138	3 to 8 decoder w/active low enable, inv out	11
T139	2 to 4 decoder w/enable, inv out	4
T151	8 to 1 MUX w/ enable, true and inv out	5
T153	4 to 1 MUX w/ active low enable	2
T157	2 to 1 MUX w/enable	1
T161A	4 bit counter	21
T164	8 bit parallel out serial shift register	17
T181	4 bit alu	31
T194	4 bit shift register	14
T195	4 bit shift register	10
T273	octal D ff w/clear	16
T269	8 bit up/down counter	80
T280	parity generator/checker	9
T377	octal D ff w/active low enable	16
Decoders:		
DEC2X4	2 to 4 decoder	4
DEC2X4A	2 to 4 decoder, inv out	4
DECE2X4	2 to 4 decoder w/enable	4
DECE2X4A	2 to 4 decoder w/enable, inv out	4
DEC3X8	3 to 8 decoder w/enable	8
DEC3X8A	3 to 8 decoder w/ enable, inv out	9
DECE3X8	3 to 8 decoder w/enable	11
DECE3X8A	3 to 8 decoder w/enable, inv out	11
DEC4X16A	4 to 16 decoder, inv out	20
Fast Adders:		
FADD8	8 bit fast adder	37
FADD12	12 bit fast adder	58
FADD16	16 bit fast adder	78
FADD24	24 bit fast adder	120
FADD32	32 bit fast adder	160
Multiplexers:		
MX8	8 to 1 MUX	3
MX8A	8 to 1 MUX, inv out	3
MX16	16 to 1 MUX	5

Actel Soft Macros, cont.

Macro Name	Description	Module Count
Registers:		
REG8	octal register w/ preset & clear	16
REG8A	octal register, inv, clock, enable, preset, & clear	16
Identity Comparators:		
CMP4	4 bit identity comparator	5
CMP8	8 bit identity comparator	9
CMP16	16 bit magnitude comparator	93
Magnitude Comparators		
MCMP2	2 bit Mag comparator w/ enables	9
MCMP4	4 bit Mag comparator w/ enables	18
MCMP8	8 bit Mag comparator w/ enables	36
MCMP16	16 bit magnitude comparator	93
Latches:		
LAT8	octal latch w/enable	8
LATM8	octal latch, MUXed inputs	8
LAT8A	octal latch w/clear	8
Shift Registers:		
SREG4	4 bit shift register w/clear	8
SREG8	8 bit shift register w/clear	16
Counters:		
CNT4BH	4 bit binary ctr. (active high-carry out) w/ clear, preset	18
CNT4BL	4 bit loadable binary counter (active low-carry out) w/clear, preset	15
Multiplier:		
MULT8	8 X 8 twos complement multiplier	235

TTL-to-Actel Cross Reference Guide

TTL Part Number	Actel Libary Component Name
7400	NAND2
7402	NOR2
7404	INV
7408	AND2
7410	NAND3
7411	AND3
7420	NAND4
7421	AND4
7427	NOR3
7432	OR2
7474	DFPC (PRE active high)
7486	XO
74109	JKFPC (PRE active high)
74138	T138
74139	T139
74151	T151
74153	T153
74157	T157
74161A	T161A
74164	T165
74181	T181
74183	FA1B (Carry-in, Carry-out active low)
74191	T191
74194	T194
74273	T273
74269	T269
74280	T280
74377	T377
74810	XNOR

Application Notes:

Gate Array Design
UART Design

Applications Briefs:

Three-Stating ACT 1010/1020 Designs
ALU181
Fast Adders
8-bit Twos Complement Multiplier
The Actel Timer
Using the Actionprobes
Metastability

Introduction

About the Actel Application Guide for ACT 1 Gate Arrays

This Application Guide explains how to design circuits for Actel logic devices while obtaining maximum efficiency of logic module utilization and maximum performance. It also describes both desirable and undesirable (i.e., error-producing) design practices.

The first sections focus on use of the Actel macro library for efficient logic module utilization.

Next the guide describes design practices which affect both utilization efficiency and performance.

The Guide then describes the use of MUXs to implement circuits which would otherwise use three-state functions.

Finally, the Guide covers performance and routability for Actel Gate Array designs with special focus on critical paths and I/O pin assignment.

Most Efficient Actel Macros

Certain classes of Actel macros allow a maximum amount of logic function to be implemented using a minimal number of Actel logic modules.

Figure 1 below shows the most efficient one-module Actel macros; Figure 2 shows the most efficient two-module Actel Macros.

Figure 1: Most Efficient 1-Module Actel Macros

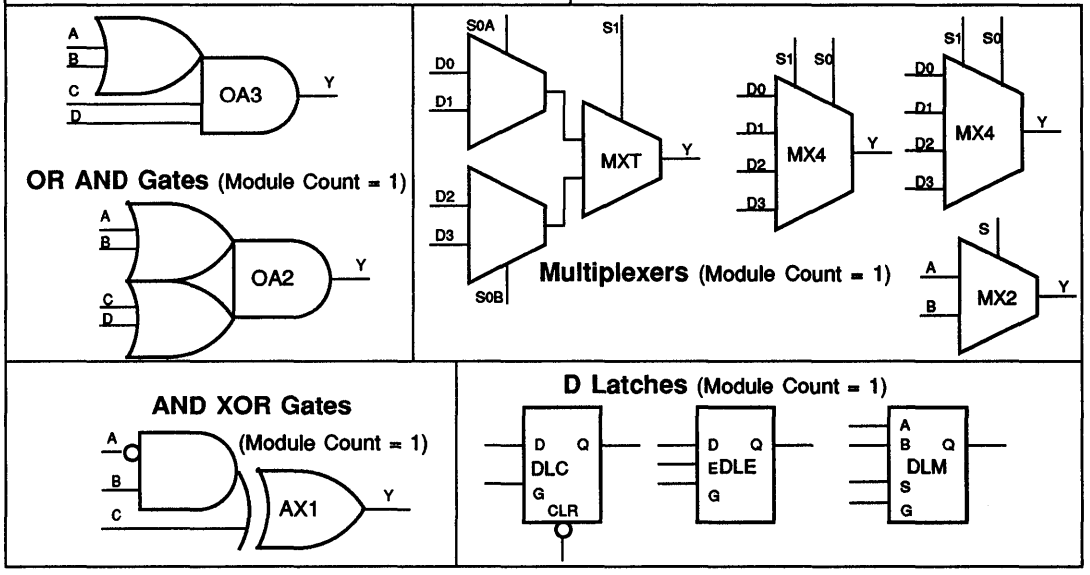
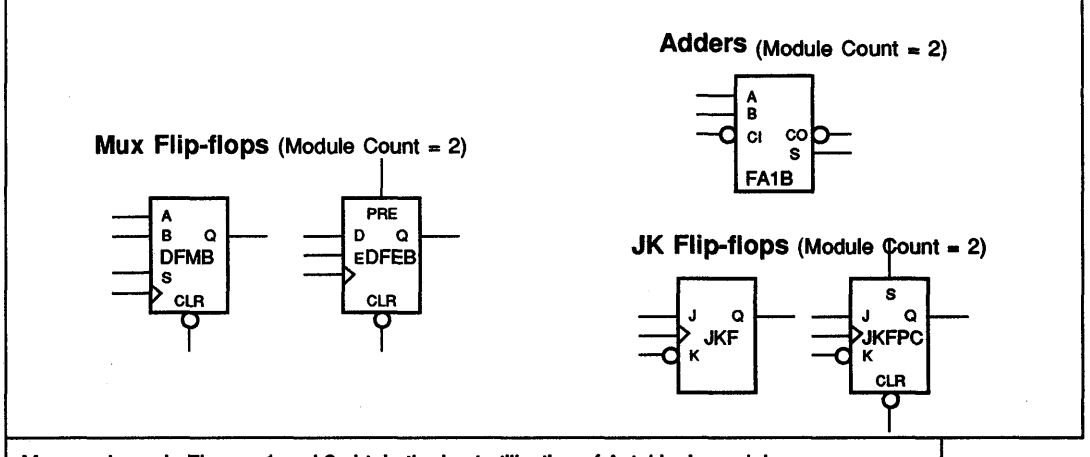


Figure 2: Most Efficient 2-Module Actel Macros



Macros shown in Figures 1 and 2 obtain the best utilization of Actel logic modules.

The Least Efficient Actel Macros

Buffers and Inverters

Buffers and inverters are the least efficient Actel macros. Designers should avoid using them except where necessary to reduce fanout. Eliminate inverters by using gates with input negation bubbles. (See the section in this guide about negation bubbles.)

Buffers are necessary to reduce fanouts to acceptable numbers, although buffers can sometimes be eliminated by using redundant logic functions.

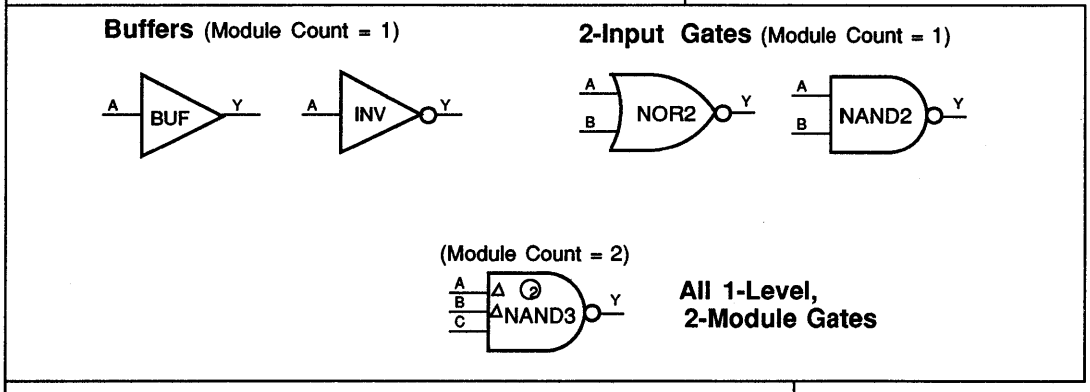
Two-Input Gates

Two-input gates are also relatively inefficient. To avoid using two-input gates, use instead MUX inputs on flip-flops, where possible, to implement two-input functions.

1-level, 2-Module Gates

Single level logic gates requiring two Actel modules are inefficient and add extra delay on some pins; avoid using these gates when possible.

Figure 3: Inefficient 1- and 2- Module Macros



Buffers, inverters, and 1-level, 2-module gates are the least efficient.

Configuring Multiplexed Inputs as AND Gates

Sometimes a simple logic function preceding a flip-flop or latch can be eliminated by incorporating the function into a flip-flop or latch having MUX input capability. See Figure 4 Below.

Configuring Multiplexed Inputs as OR Gates

Using MUX inputs on flip-flops and latches can help eliminate simple logic functions which would otherwise require an additional logic module. In this way, it is possible to fit more logic functionality into an ACT device. See Figure 5 below.

Figure 4: Configuring Multiplexed Inputs as AND Gates

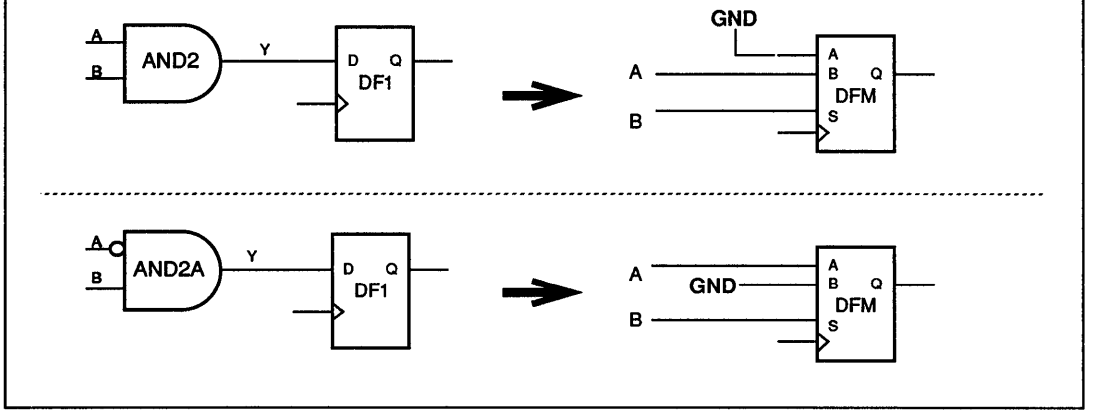
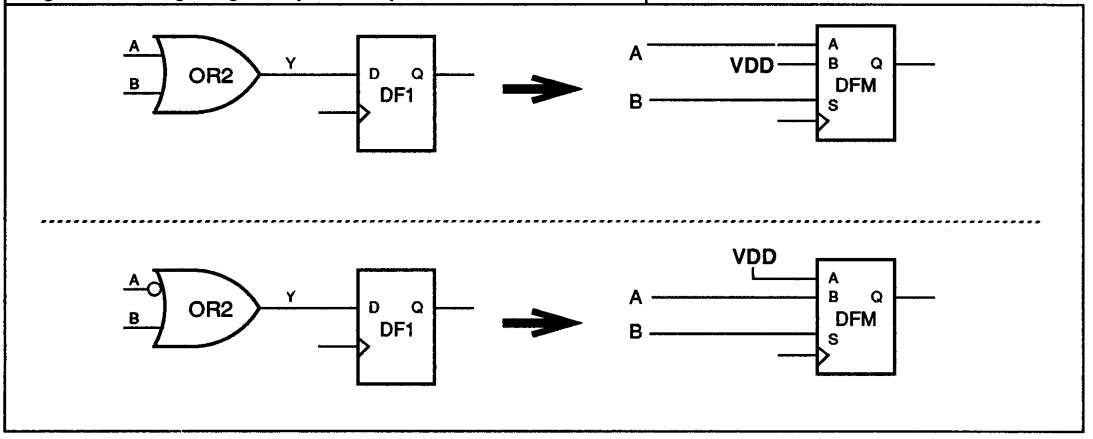


Figure 5: Configuring Multiplexed Inputs as OR Gates



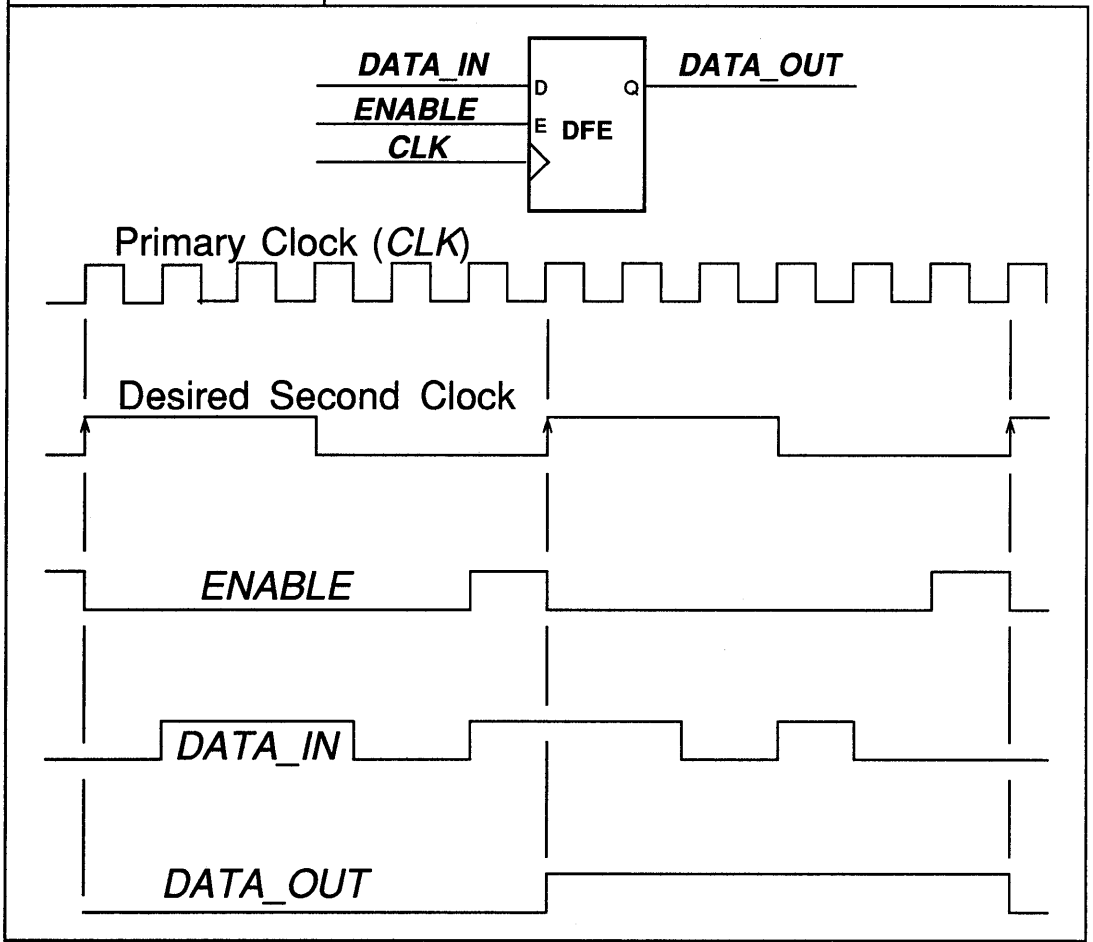
Clock Enables

Clock Enables allow data storage and transfer at different intervals while using a common clock

Common clock, synchronous systems work best. These systems are the easiest to design, debug, and analyze for timing constraints.

When it is necessary to store data at a time interval differing considerably from the primary clock frequency, it may seem desirable to introduce a second clock. Usually, however, the designer may avoid this second clock by using flip-flops or latches with an Enable function. See Figure 6 below.

Figure 6: Clock Enables



Avoid using a second clock by using flip-flops or latches with an Enable function.

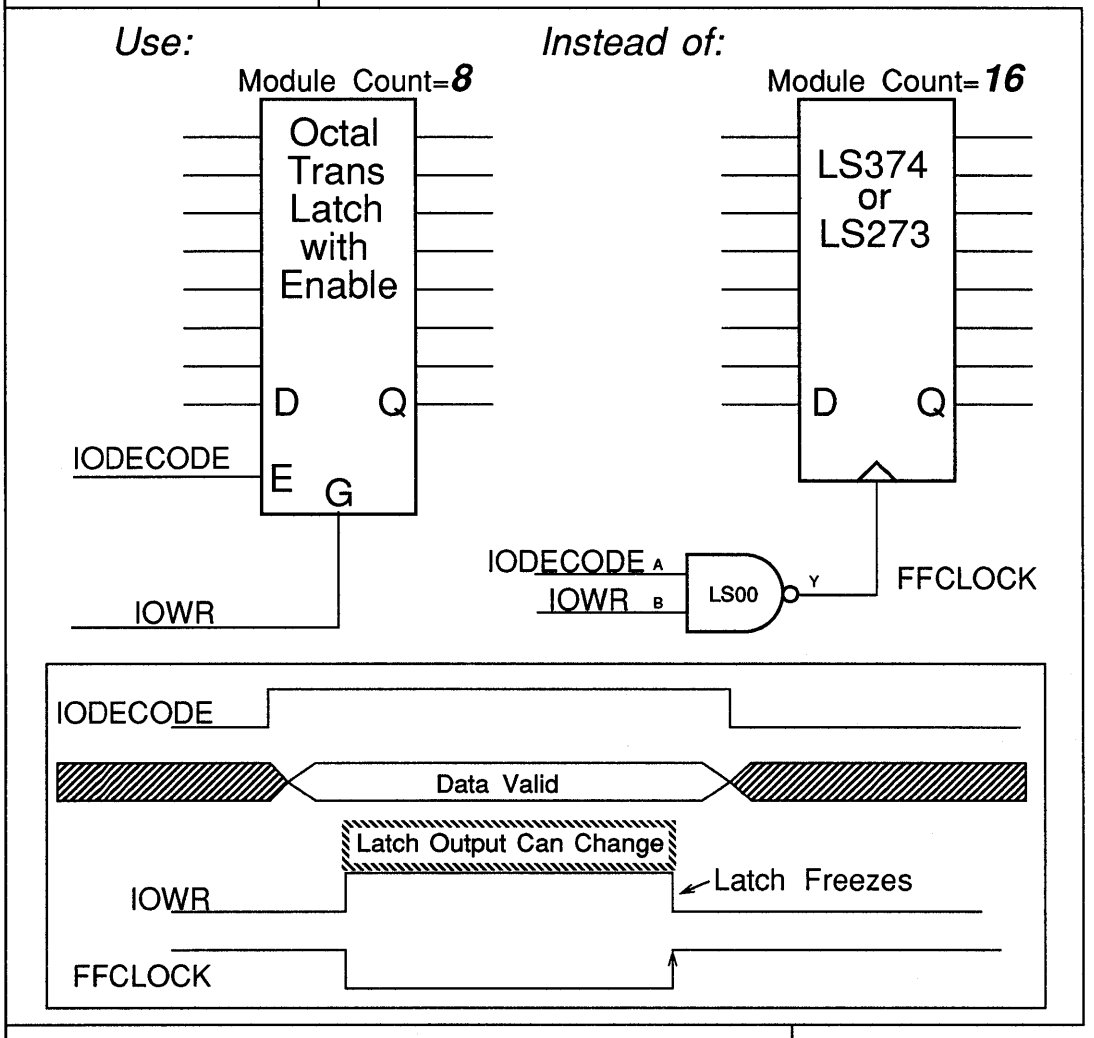
Efficient Data Storage

Where possible, use transparent latches instead of D flip-flops for data storage, because they use only one logic module (flip-flops use two logic modules).

For example, with a data register in a typical microprocessor system, if a simple data storage function is possible using a transparent latch *or* a flip-flop, use the latch instead. See Figure 8 below.

Since data storage may require many elements, the additive advantage of using latches instead of flip-flops is often significant in the overall completed design.

Figure 8: Data Storage



Use latches instead of flip-flops whenever possible.

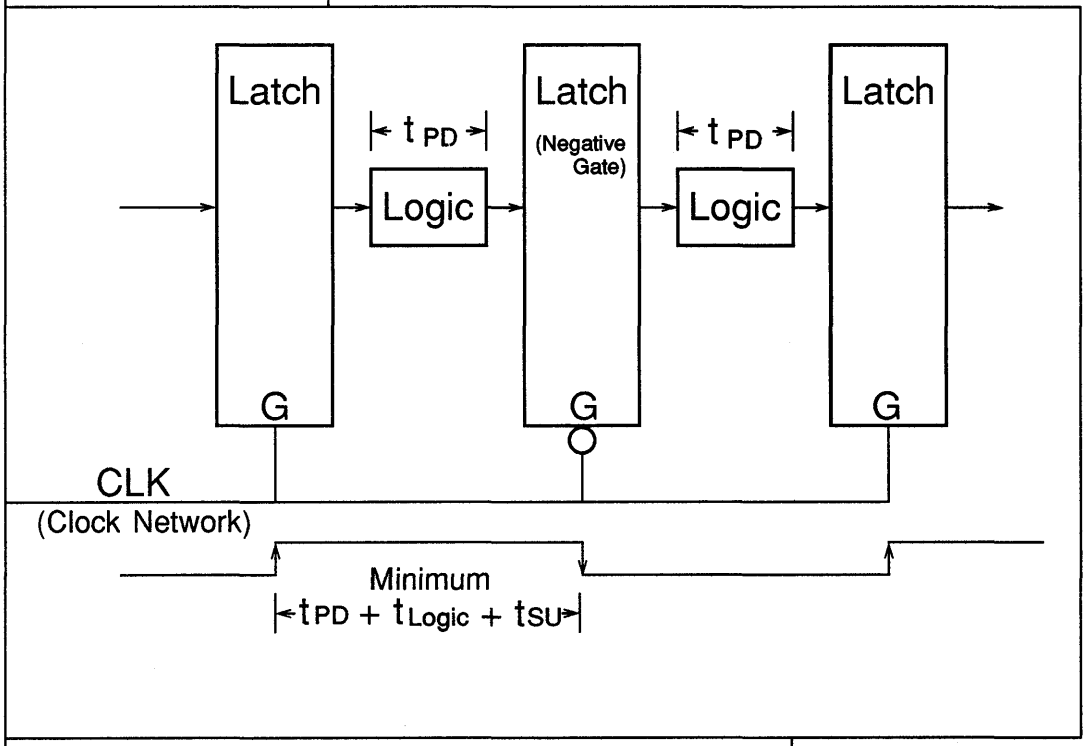
Pipelined Logic Using Transparent Latches

In addition to transferring pipelined data, transparent latches may also be used for pipelined logic where the timing considerations are as shown in Figure 9.

To facilitate pipelined applications using transparent latches, the Actel macro library includes both negative and positive gates on latch macros; this facilitates the use of the clock's opposite phases.

This design practice has the same effect as a two-phase clock system, yet uses only the single clock network found in ACT devices.

Figure 9: Pipelined Logic



Use transparent latches to pipeline logic.

Negation Bubbles

TTL-based system designs generally have numerous inverters (usually LS04s) throughout the design. Most designs implemented in conventional gate arrays also extensively use inverters .

In contrast to TTL-based systems and conventional gate-arrays, designs implemented on ACT devices use fewer inverter macros due to a unique feature of the Actel macro library: negation bubbles on macro inputs.

In addition to gate structures found in most gate array libraries, the Actel macro library includes structures with inversion bubbles, in different combinations, on logic module inputs. Taking advantage of these macros helps minimize the number of logic modules required to implement your logic function. Figures 11, 12 and 13 on the next page show how input negation makes logic more effective.

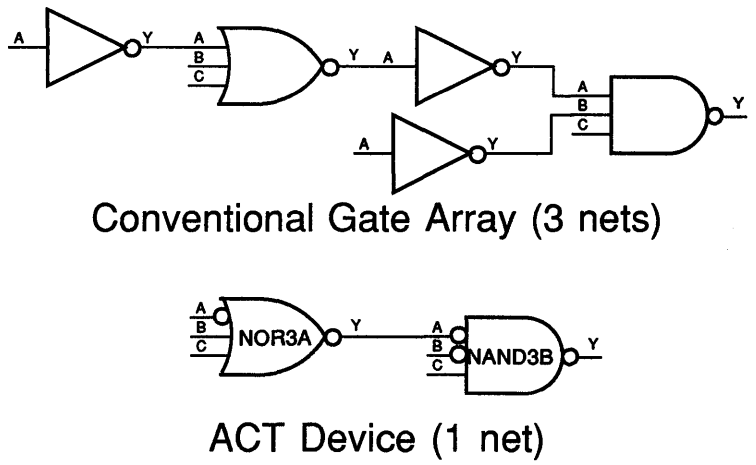
Potentially more important is the ***INCREASE IN PERFORMANCE*** made possible using fewer logic module and wiring delays in paths. Figure 15 below compares an ACT design and a gate array implementing the same function.

In this example, the conventional gate array has four macro delays, while the ACT design has only two delays.

Figures 16-18 on page 15 show how different families of Actel macros contain different combinations of input negation bubbles.

Figures 14 and 15 on page 11 show a critical path from a 16-bit synchronous counter example, implemented both in a conventional gate array, and in Actel macros. Note that the ACT design requires only six nets in the path, compared to 11 nets for the gate array.

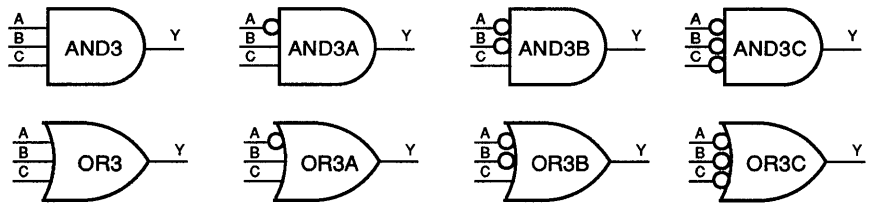
Figure 10: Negation Bubbles



Negation bubbles on macro inputs allow fewer levels of logic to be utilized within the design.

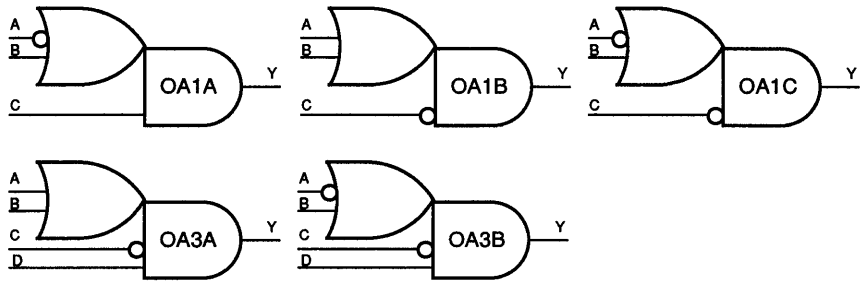
Negation Bubbles on Macro Inputs

Figure 11: Logic Combinations



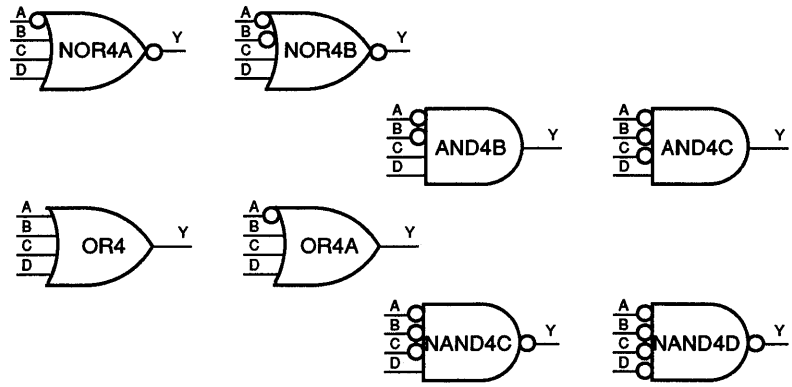
Some gate families have all combinations available.

Figure 12: Two-Level Logic



Two-level logic is more effective with input negation

Figure 13: One Level Logic with Input Negation



These input negation possibilities are available on 4-input, 1-module gates.

ACT Device vs. Conventional Gate Array: 16-Bit Synchronous Counter Comparison

Figure 14: Critical Path for ACT Device (6 nets)

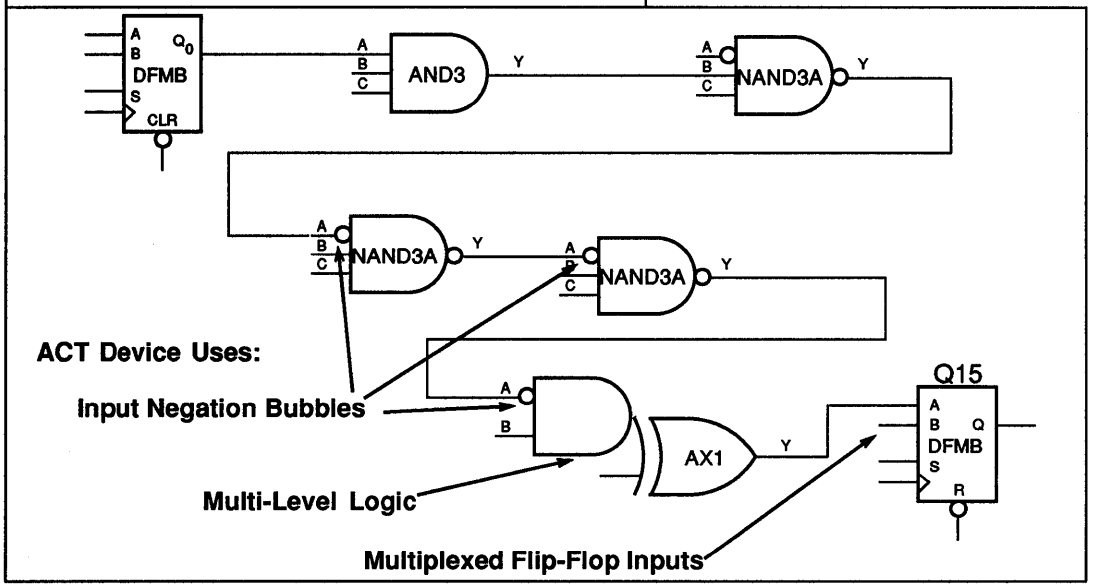
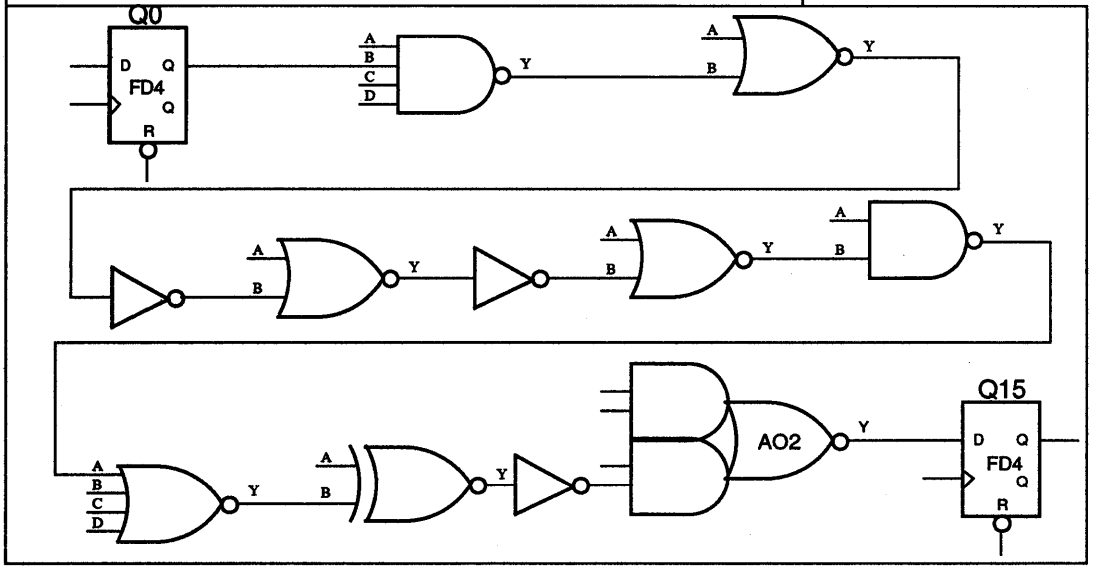


Figure 15: Critical Path for Conventional Gate Arrays (12 nets)



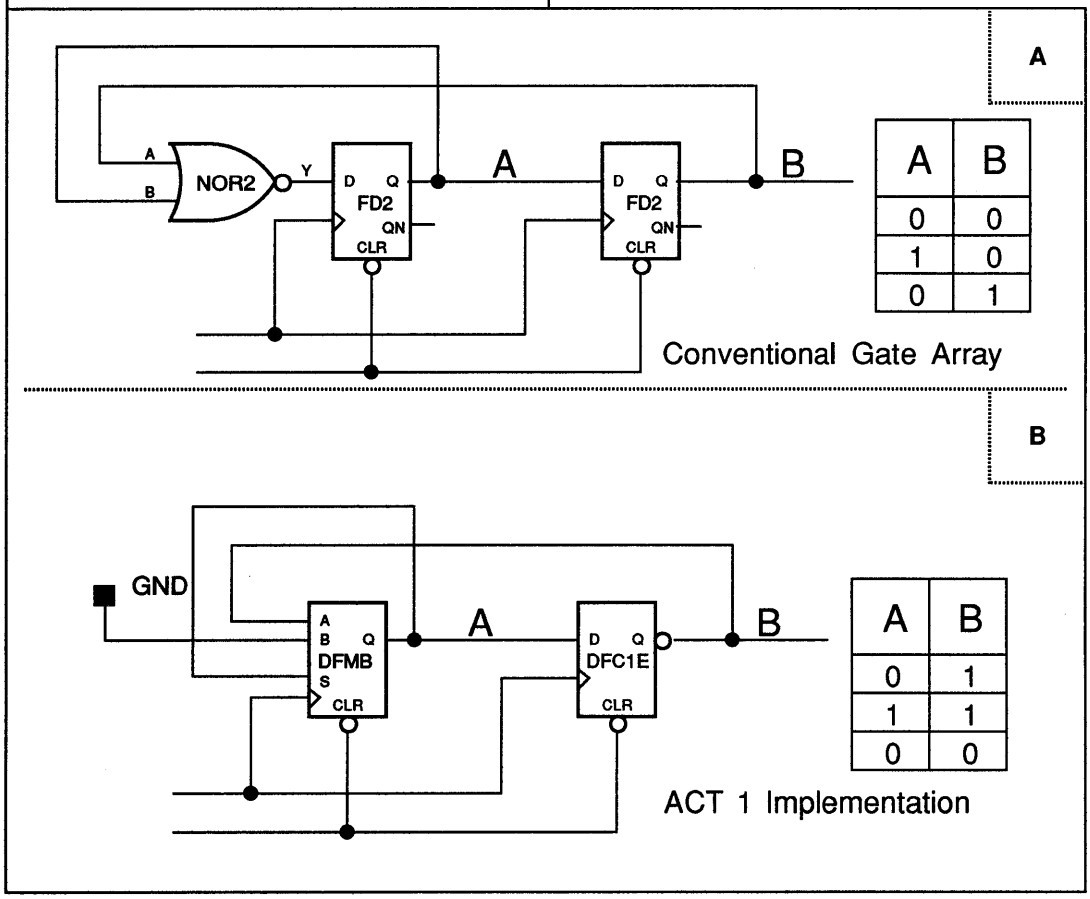
High-Speed Prescalers/Counters

It is often necessary to pre-scale an incoming signal or create a pattern of waveforms, using a very high speed synchronous counter. This is possible in the ACT 1 architecture using the logic functions of input MUXs on certain flip-flop macros.

Figure 16 below shows a Divide-By-3 counter (also called a modulo 3 counter) implemented in two ways. Part A of Figure 16 shows a conventional masked-programmed gate array requiring a separate NOR gate, in addition to the two flip-flop macros. Part B of Figure 16 shows effective use of the ACT library: the modified NOR function fits one AND configuration, made possible by the MUX input. The second flip-flop now has an inverting output suited to the gate function of the MUX input on the first flip-flop.

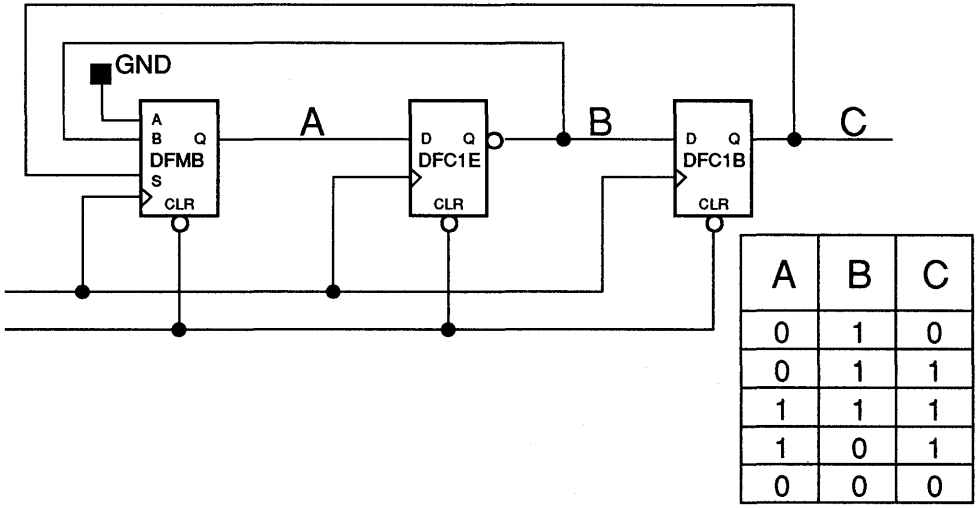
The state table for the counter has changed, but it still implements a divide-by-3 function without additional delay between flip-flop outputs and inputs.

Figure 16: Divide-By-3 Counter/Prescaler



Similar principles were applied in creating the divide-by-5 counter/prescaler shown below in Figure 17. When using circuits like these to create a widely-used, lower frequency clock, it may be necessary to attach a prescaler output to the ACT clock network dedicated I/O pin. If this is true, be careful to place the prescaler input and output I/O pins close to the dedicated clock pin. Also, watch the delays on the net which drives the prescaler clock; it will be more prone to clock skew than the dedicated clock network.

Figure 17: Divide-By-5 Counter/Prescaler Using ACT Library



A	B	C
0	1	0
0	1	1
1	1	1
1	0	1
0	0	0

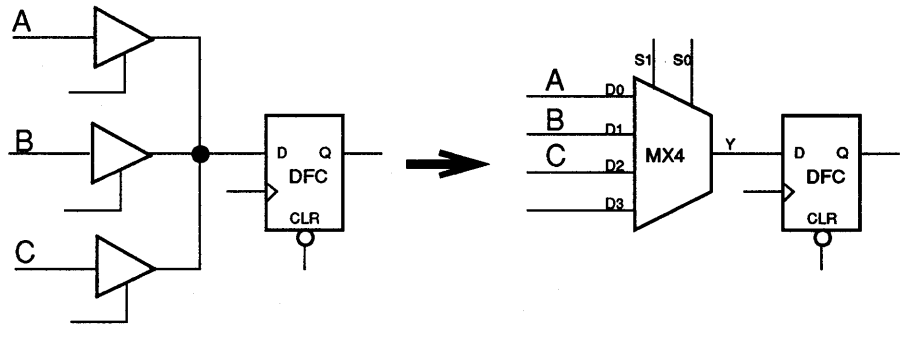
Use MUXs Instead of Three-State Buffers in ACT devices

ACT devices do not have true three-state functionality for circuits implemented within the device. However, it is easy to implement all three-state functions using MUXs. See Figure 18 below.

Since MUX-type structures are the most efficient of all Actel macros, using MUXs produces an added performance benefit (MUXs are always faster than three-state alternatives). Using MUXs also removes the problem of bus conflicts.

Use MUX inputs on flip-flops, when possible, to gain an efficiency and performance advantage.

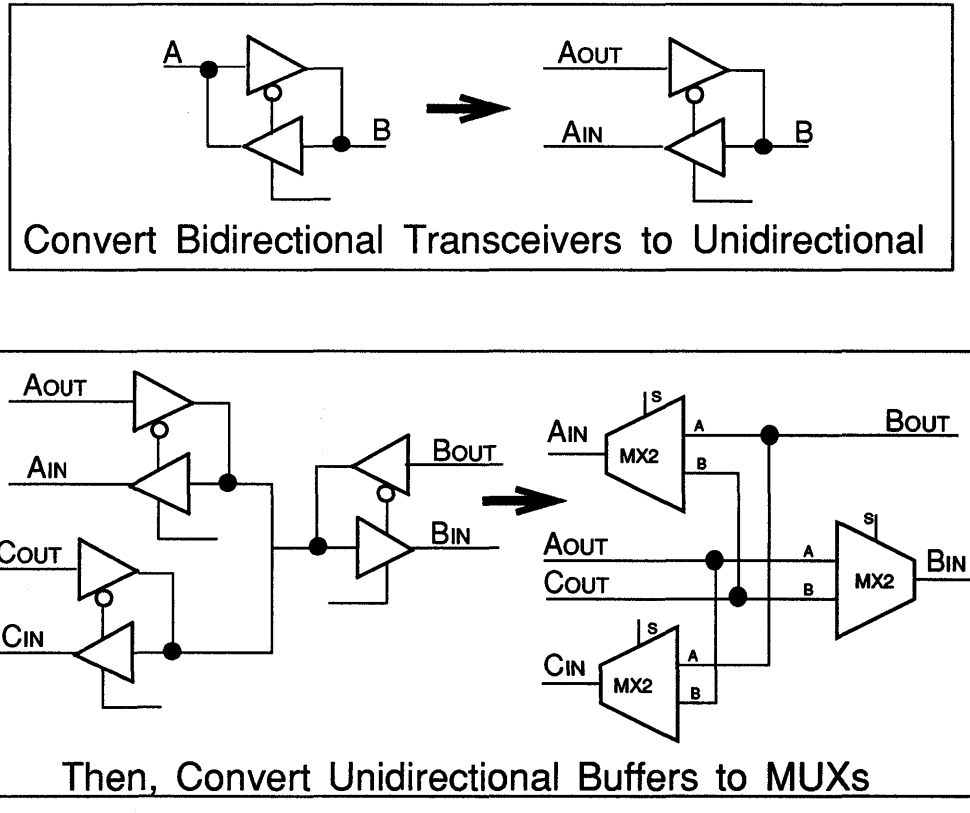
Figure 18: Using MUXs for Three-State Functions



Internal Three-state Transceiver Functions

When implementing internal three-state transceiver functions in ACT devices connecting multiple bus structures, the process is easier if performed in two steps as shown in Figure 19. First break bi-directional structures into two unidirectional structures. It is then simple to complete conversion to a MUX-based circuit.

Figure 19: Implementing Bus Transceivers within ACT Devices



Designing for Performance and Routability

Modification of *fanout* and *I/O assignment* may significantly improve the performance and routability of your design. Fanout considerations are addressed below and on the next page; I/O assignment is discussed on page GA-18.

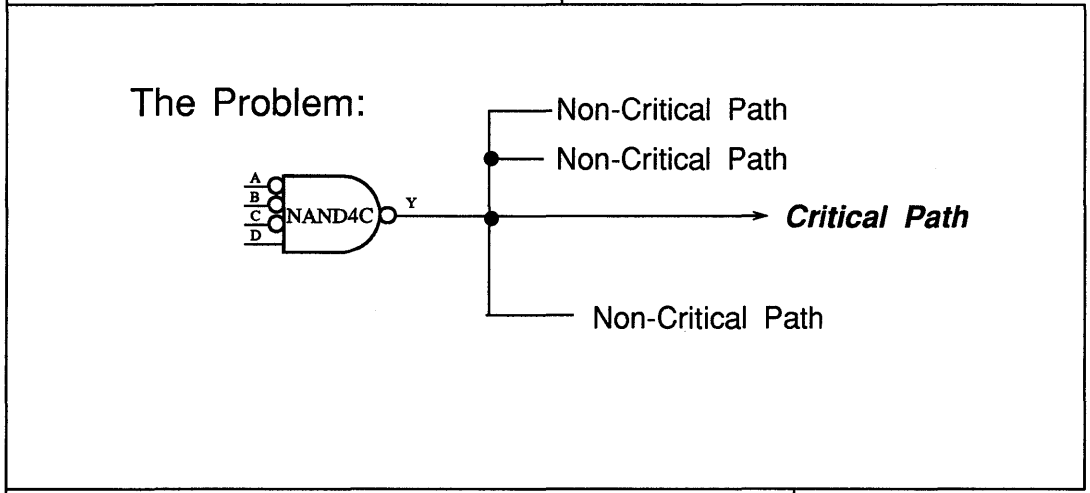
Designing Fanout for Optimized Performance and Routability

Consider a net in one of the design's critical paths, illustrated in Figure 20 below.

Of all input pins on the net, only the input pin in the critical path is important to optimizing the design's performance. The other input pins on the net create additional loading, slowing signals propagating along the critical path.

Excessive fanouts also create longer, more complex nets, causing routing congestion and limiting the design's routability. This reduces the ACT device's logic capacity for the desired application.

Figure 20: High fanouts Slow Critical Paths



Many nets containing critical paths also include portions of non-critical paths.

To solve this problem, isolate the critical path from the non-critical paths, as shown on page 19, without disturbing the logical function of the circuit.

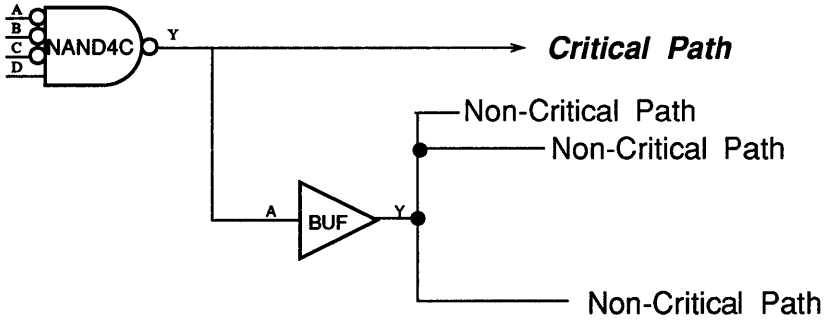
There are two solutions to the problem of high fanout on a critical path. Solution A buffers the non-critical paths to reduce fanout on the critical path.

When the added buffer delay is too detrimental to the non-critical paths, Solution B (redundant logic) is an alternative. Solution B requires that the added loading on input nets does not cause additional timing problems.

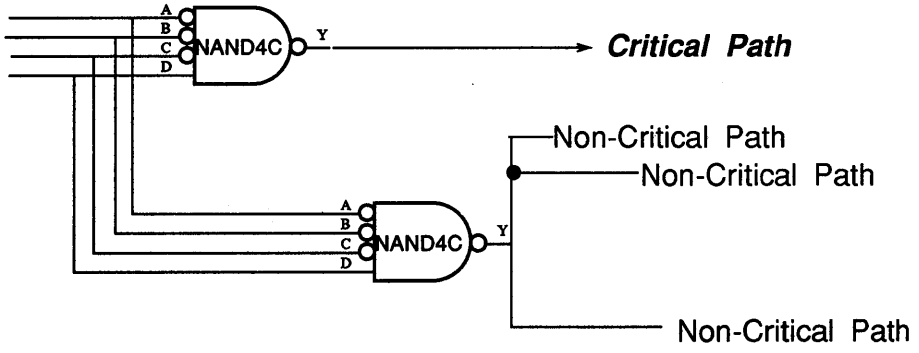
Another way to improve critical net performance is to label the net "critical" causing the Route program to optimize the routing of the net. However, to achieve the greatest performance improvements, both careful isolation of critical paths and careful choice of I/O pin assignment are necessary.

Figure 21: Two Solutions for Improving Performance

Solution A: Buffering:



Solution B: Redundant Drive:



Assigning I/O Pins
For Optimized Performance and Routability

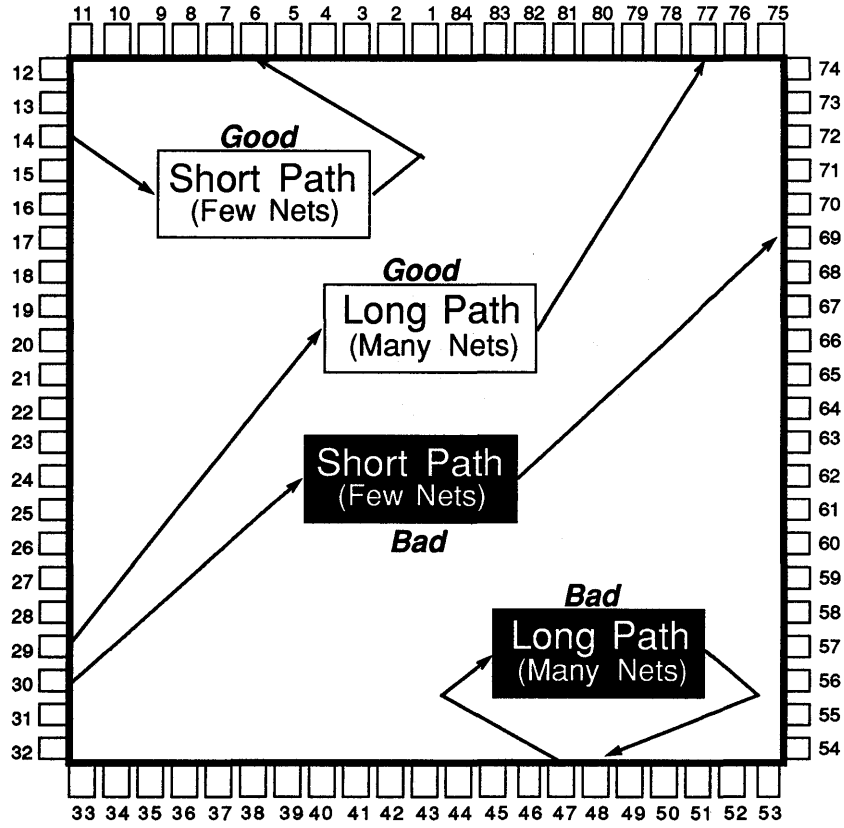
When the Placement program assigns module locations to implement Actel hard macros, it evaluates the I/O pin locations to optimize I/O placement.

Figure 22 shows poor I/O pin assignment may cause excessively long nets if the I/Os of a short logic path are too far apart.

For long logic paths, (i.e., paths containing many levels of logic), poorly placed I/O pins may also cause long nets if they are too close together.

Long nets add routing congestion, limiting the design's capacity, and adding additional delay.

Figure 22: Optimizing I/O Pins Assignment

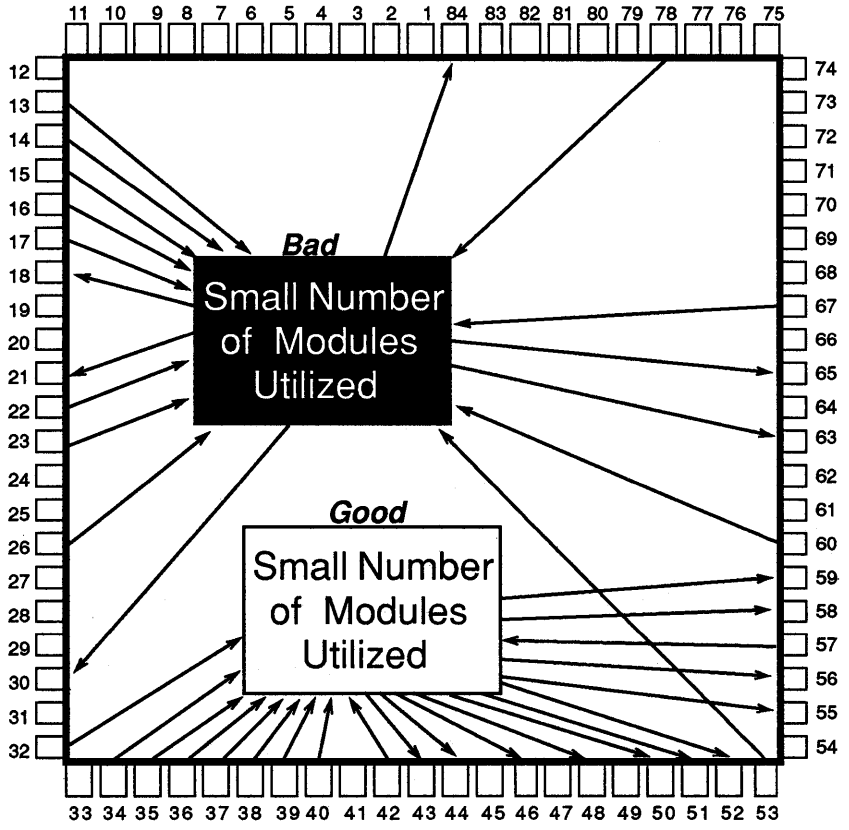


Bad I/O assignment may cause excessively long nets.

For optimum design performance, pay careful attention to I/O pin assignment. For example, if I/Os are "scattered" around the package periphery, a design with low module utilization will likely have long nets with excessive delays.

When assigning I/O pins for designs utilizing a relatively small percentage (<50%) of the available logic modules, it is preferable to assign I/O pins so that they are relatively close together. See Figure 23, below.

Figure 23: Optimizing I/O Pin Assignment



Introduction

About the Actel Application Note for UART Design

This Application Note demonstrates the design of a complex circuit using Actel desktop configurable gate arrays. The Note will show:

1. the flexibility of ACT 1 family architecture;
2. features of the ACT 1 macro library which speed and simplify the design process;
3. the Actel design process using Viewlogic design entry and the Action Logic™ System;
4. how to design a Universal Asynchronous Receiver & Transmitter (UART) with the Actel Design Process.

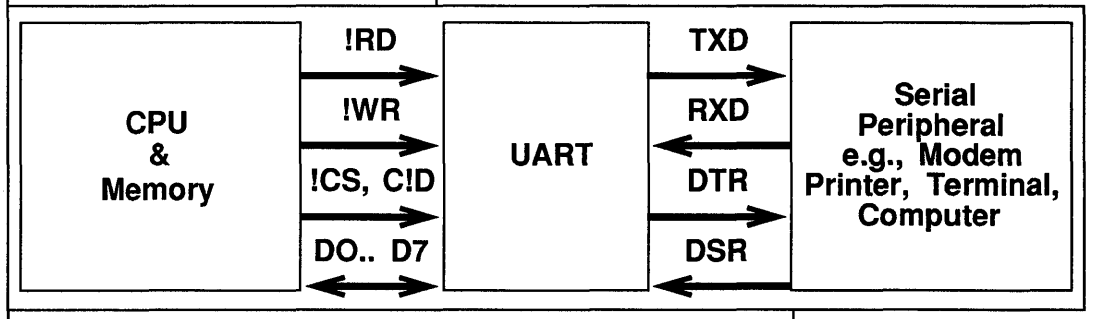
UART Description

The UART is a serial communication device which handles transmission and reception of information between computers and peripherals including modems, terminals and printers. This Note describes a general purpose UART which utilizes 178 logic modules in the ACT 1010, representing approximately 700 gates.

Figure 2 below shows the UART configured as a controller between a computer and a peripheral device. Communications protocol is a standard RS232 serial interface.

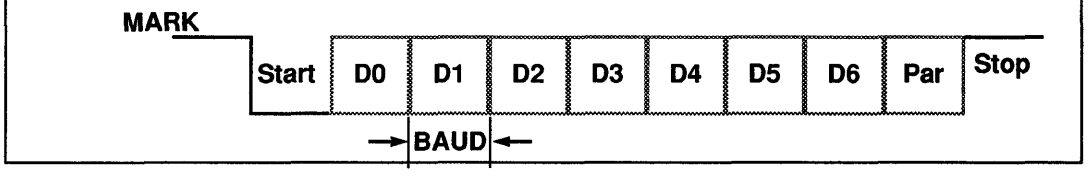
The UART transmission format is shown in Figure 3 below. A start bit begins the transmission, followed by the data bits, and then an optional even or odd parity bit. A programmed number of stop bits concludes the transmission.

Figure 2: UART System Block Diagram



The UART is configured between a CPU and a peripheral.

Figure 3: UART Transmission Format



The UART format is an RS232 serial interface.

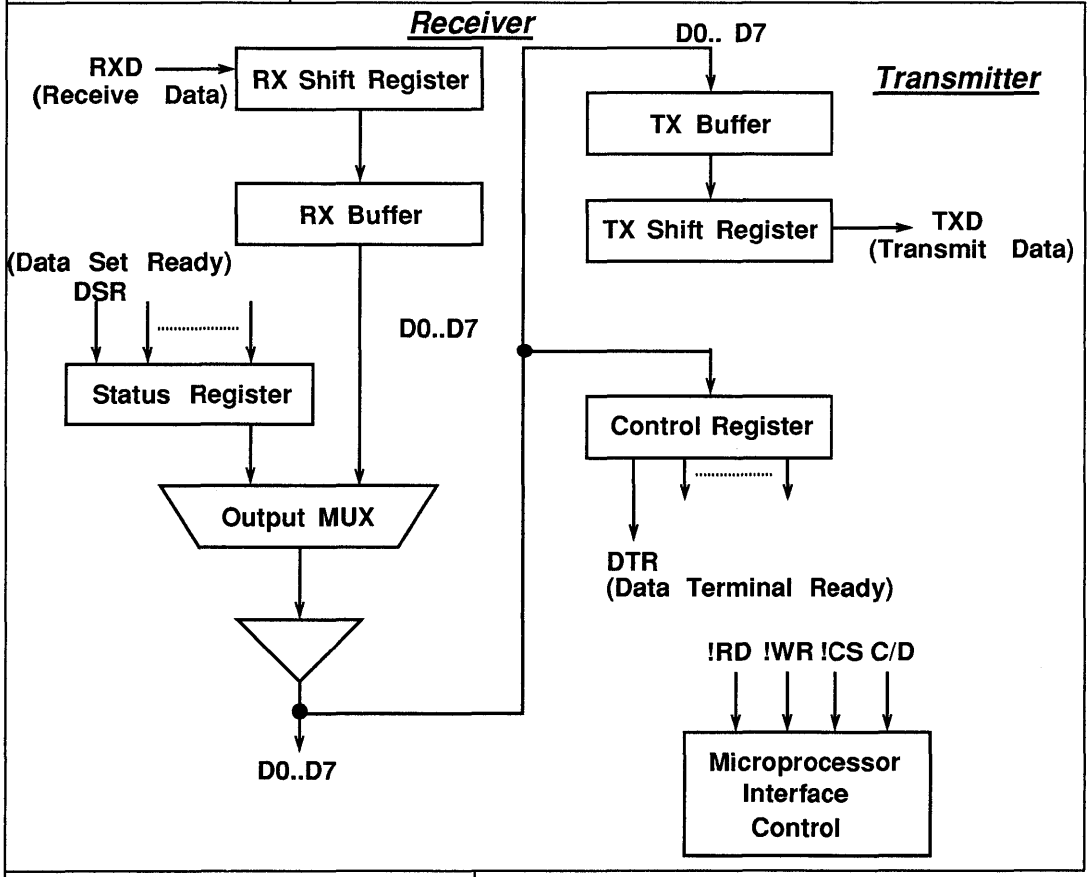
UART Design

The UART application described in this Note is similar to the INTEL 8251 operating asynchronously. See Figure 4 below.

When the CPU sends a data character, the UART automatically adds a start bit, followed by 7 (or 8) data bits, an even or odd parity bit, and a stop bit. The UART then transmits this "packet" as a serial data stream on the TXD output. It shifts out serial data a rate equal to 1/16 the clock frequency (pin 16XBAUDCLK).

When the CPU receives a data character, a falling edge on the UART's RXD input triggers the beginning of a start bit. The start bit is again checked to prevent a false start. If the start bit is valid, the data character is shifted in, followed by one stop bit. A serial stream of data is then converted to parallel format, ready for CPU retrieval.

Figure 4: UART Function

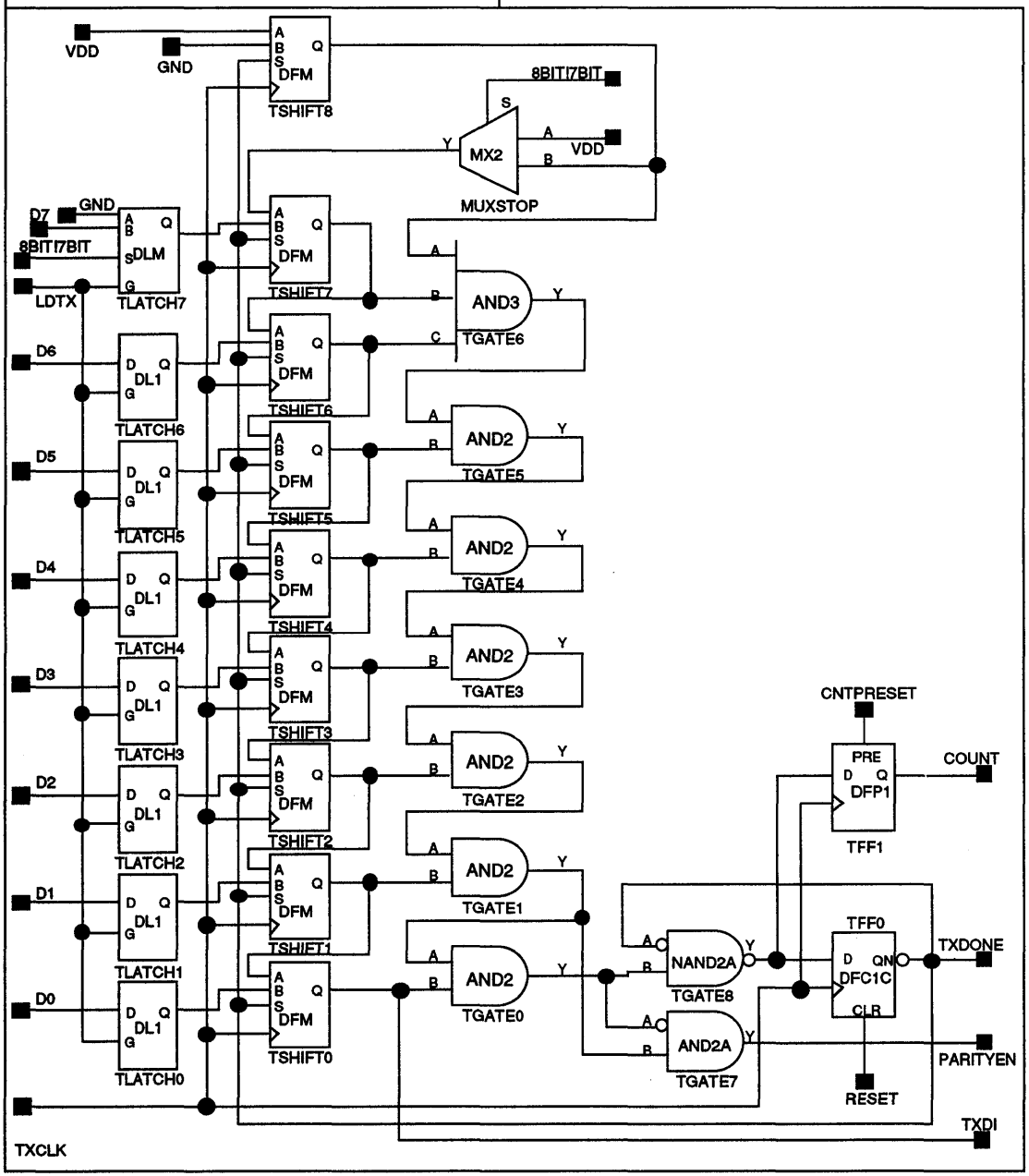


This UART application is similar to the INTEL 8251

Transmitter Design

Figure 5 below shows the transmitter's buffer and shift register.

Figure 5: Transmitter Buffer and Shift Register



The transmitter first latches the parallel data word D[0:7] from the CPU in the buffer TLATCH[0:7]. It then transfers the data character to the shift register TSHIFT[0:7] through the B inputs. After the first clock (TXCLK), TXDONE goes low selecting the A inputs of TSHIFT[0:7] . TXCLK then serially shifts out the data character. A series of AND gates (TGATE0-TGATE6) detect a logic high, indicating the stop bit, propagating through the shift register.

Transmitter Buffer

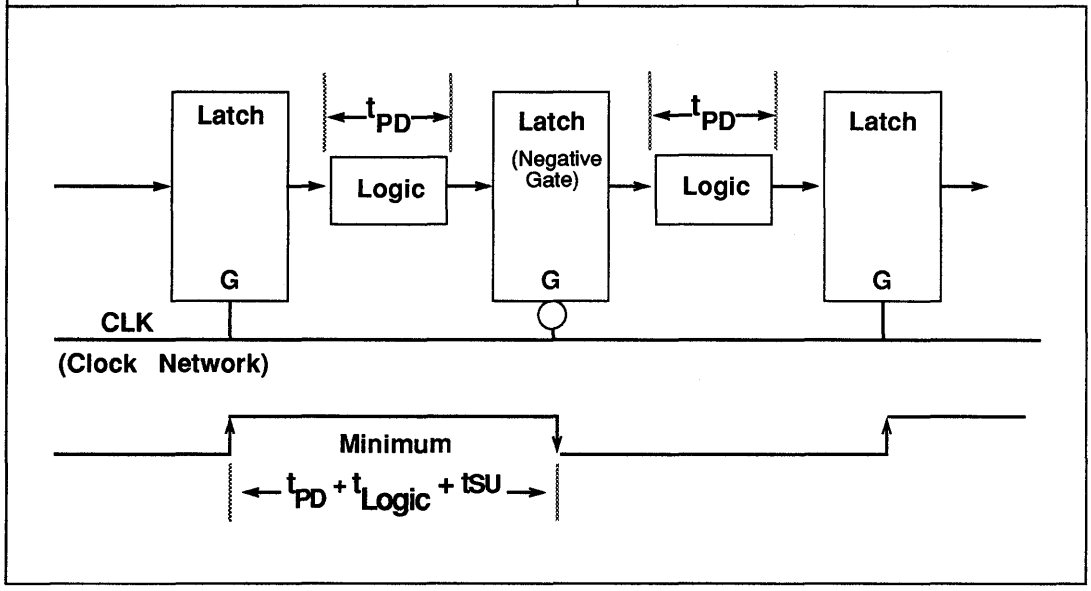
The transmitter buffer TLATCH[0-7] uses transparent latches to store a data character. The level-sensitive transparent latch consumes only one ACT 1 logic module.

Because of the wide variety of latches available in the ACT 1 library, designers may build efficient registers for any application: synchronous clear, gate-enable, active high or low outputs, and active high or low gates.

Clocks

It is also possible to construct a two-phase clock system by combining active high gate latches and active low gate latches. See Figure 6 below.

Figure 6: 2-Phase Clock Using Transparent Latches



The ACT 1 macro library includes a large selection of edge-sensitive storage elements (flip-flops, or FF's). FFs are available with asynchronous clear, asynchronous preset and clock enable inputs. Negation bubbles are available on all inputs and outputs.

The clock enable feature allows separation of data storage and transfer while maintaining a common clock system. See Figure 7, below.

Transmitter Shift Register

The transmitter's shift register TSHIFT[0:7] uses flip-flops with a built-in 2:1 multiplexor (MUX) on the D input. The built-in MUX can emulate 2-input logic functions as shown in Figure 8 below.

The MUX flip-flop uses two logic modules (as does a standard flip-flop). It is thus possible to construct a parallel load/serial shift register without additional logic.

Figure 7: Clock Enable Function

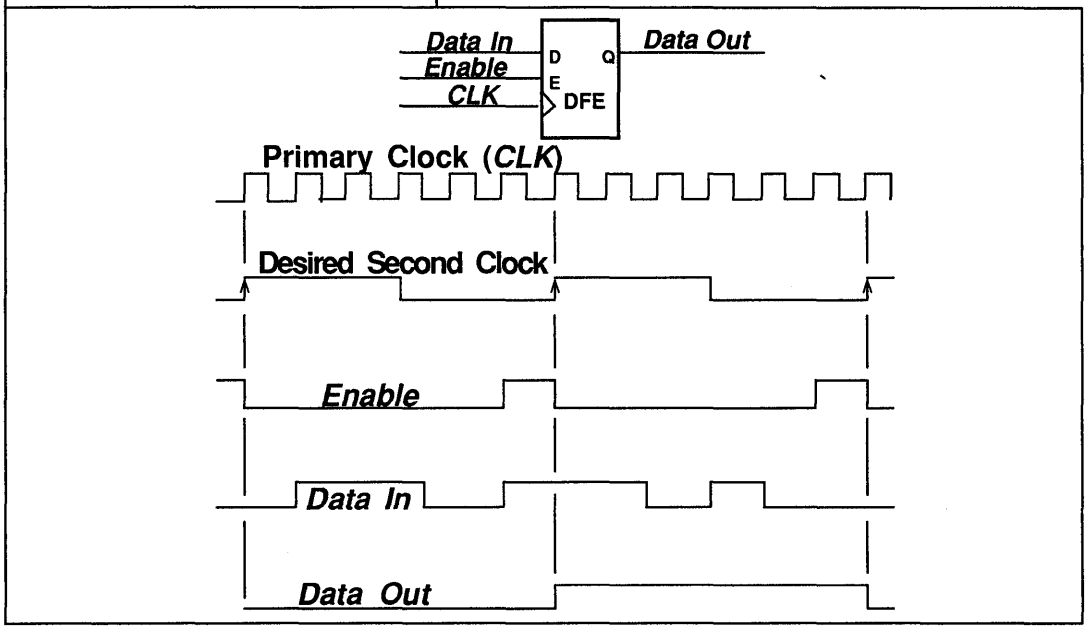
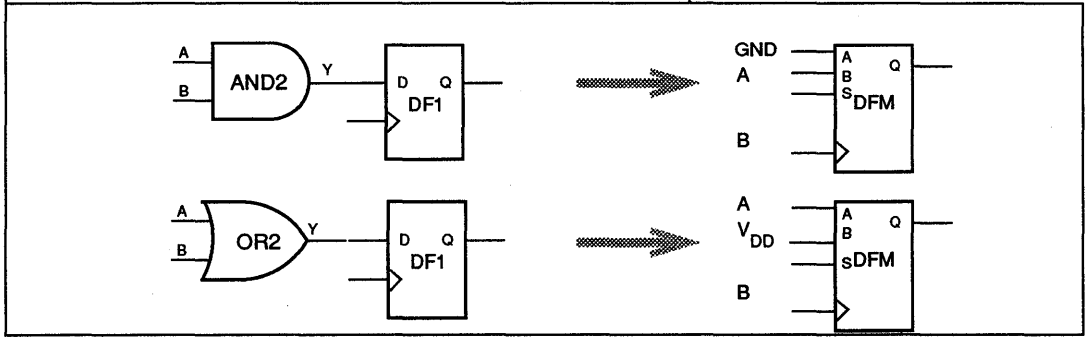


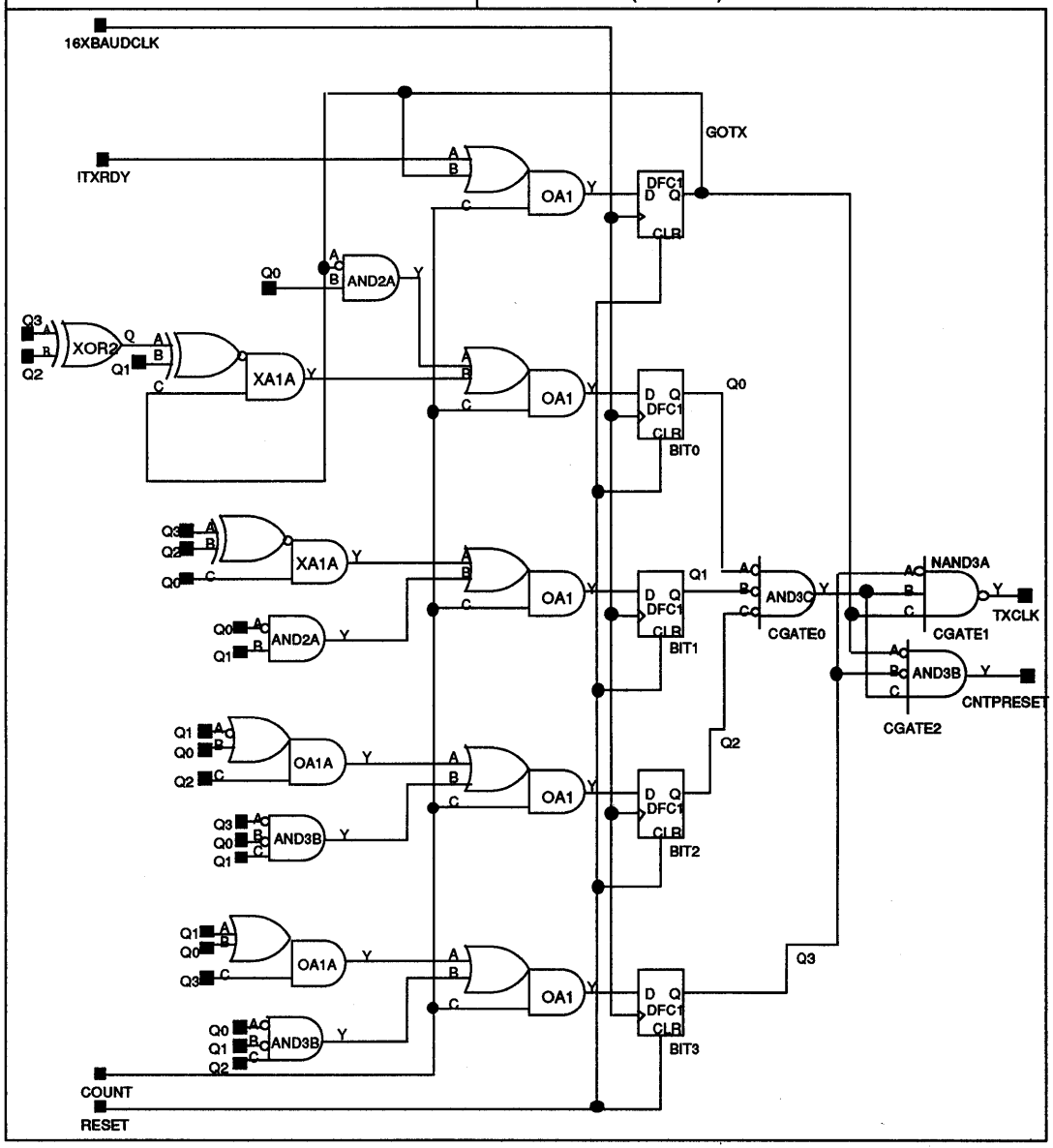
Figure 8: Reducing Logic Levels with MUX Flip-Flops



Transmitter Clock Circuit

The clock circuit is shown below in Figure 9. It is implemented as a 4-bit Gray code counter with both synchronous and asynchronous reset (COUNT & RESET, respectively) and count enable (GOTX). An external clock signal (16XBAUDCLK) drives the state machine. Two AND gates CGATE0 & CGATE1 decode the outputs Q3-Q0 of the state machine and generate an internal clock (TXCLK) for the transmitter shift register. Only one bit changes at a time, eliminating glitches on the transmitter clock (TXCLK).

Figure 9:Transmitter Clock Circuit



TWO-Level Logic

The transmitter state machine takes advantage of the OR/AND macro, a two-level logic function using only one logic module. In combinatorial logic, use of two-level logic functions can significantly reduce the module count. There are different types of OR/AND macros with a wide choice of input combinations. Refer to the ACT 1 family data sheet for a list of OR/AND and AND/OR macros.

Decoders

Decoder functions require signals and their complements. Using negation assertion levels available on Actel macros simplifies the logic path and conserves module usage. See the 2:4 decoder in Figure 10, below.

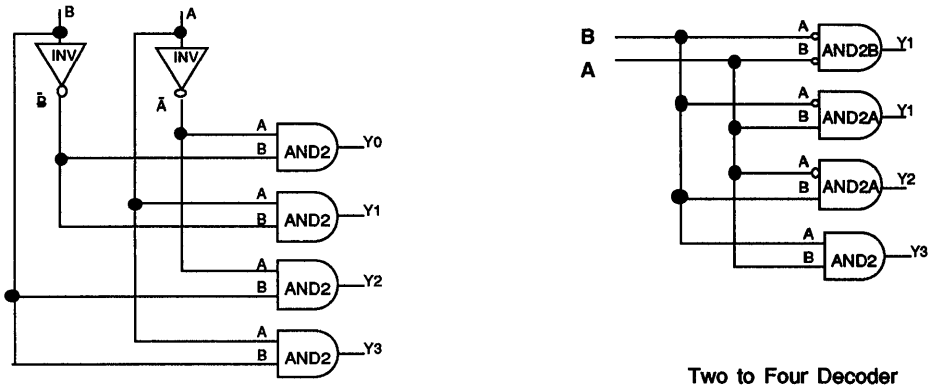
It is important to note that propagation delay through a logic module is independent of the implemented function. Thus the designer should use negation bubbles and two level logic macros whenever possible.

Parity Generation

The transmitter parity generator is programmable for odd or even parity. See Figure 11 below.

From the shift register, data serially enters the parity generator on the transmit line (TXDI). The generator inserts the parity bit just prior to the stop bit. A 2:1 MUX (PMUX) selects either data or parity bit to the data transmit line TXD.

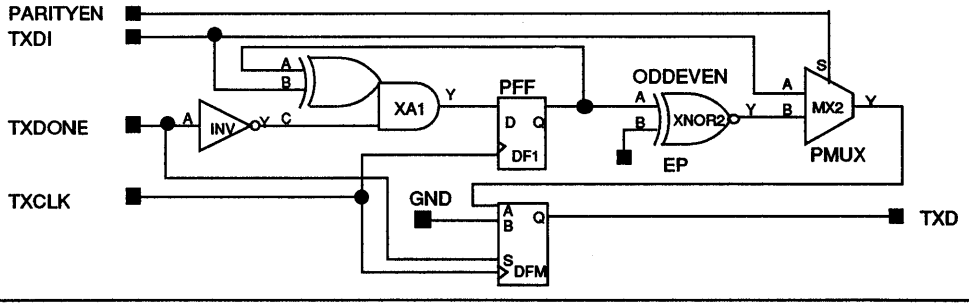
Figure 10: 2:4 Decoders



Two to Four Decoder

Two to Four Decoder
Using Actel Macro Library

Figure 11: UART Parity Generator



Logic Module Conservation

Efficient utilization of Actel macros allows implementation of the UART transmitter section using only 75 of the 295 available logic modules on the ACT 1010 array. The parity generator uses the XOR/AND macro. The XOR/AND, XOR/OR, and AND/XOR functions utilize only one logic module.

ACT 1 devices are also particularly efficient at implementing MUX functions. Both 2:1 and 4:1 MUXs can each be implemented in only one logic module. Since ACT 1 devices do not support internal three state drivers, MUXs implement internal bus designs. See Figure 12 below.

Receiver Design

Shift Register and Buffer

Figure 13 on the next page shows the schematic of the receiver's shift register and buffer. A logical "1" is loaded into the receiver shift register RSHIFT[0:9]. After the first clock (SHFTRX), XFERRX goes low, selecting the A inputs of the MUX flip-flops. The low start bit precedes the serial data shifts into the register. When a complete serial data character shifts into the register, the parallel data character transfers to the buffer (0:7) for CPU access.

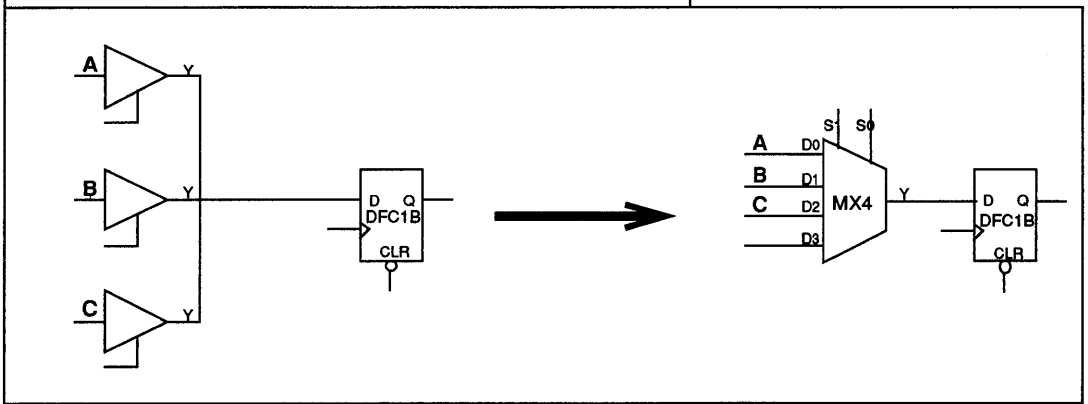
The receiver design takes advantage of the single-module transparent latch with MUX, minimizing required circuitry. Similarly, the built-in MUX can emulate other two-variable functions.

The MUX can also emulate an inverter, effecting a negation at the data input of the transparent latch.

Receiver Clock Circuit

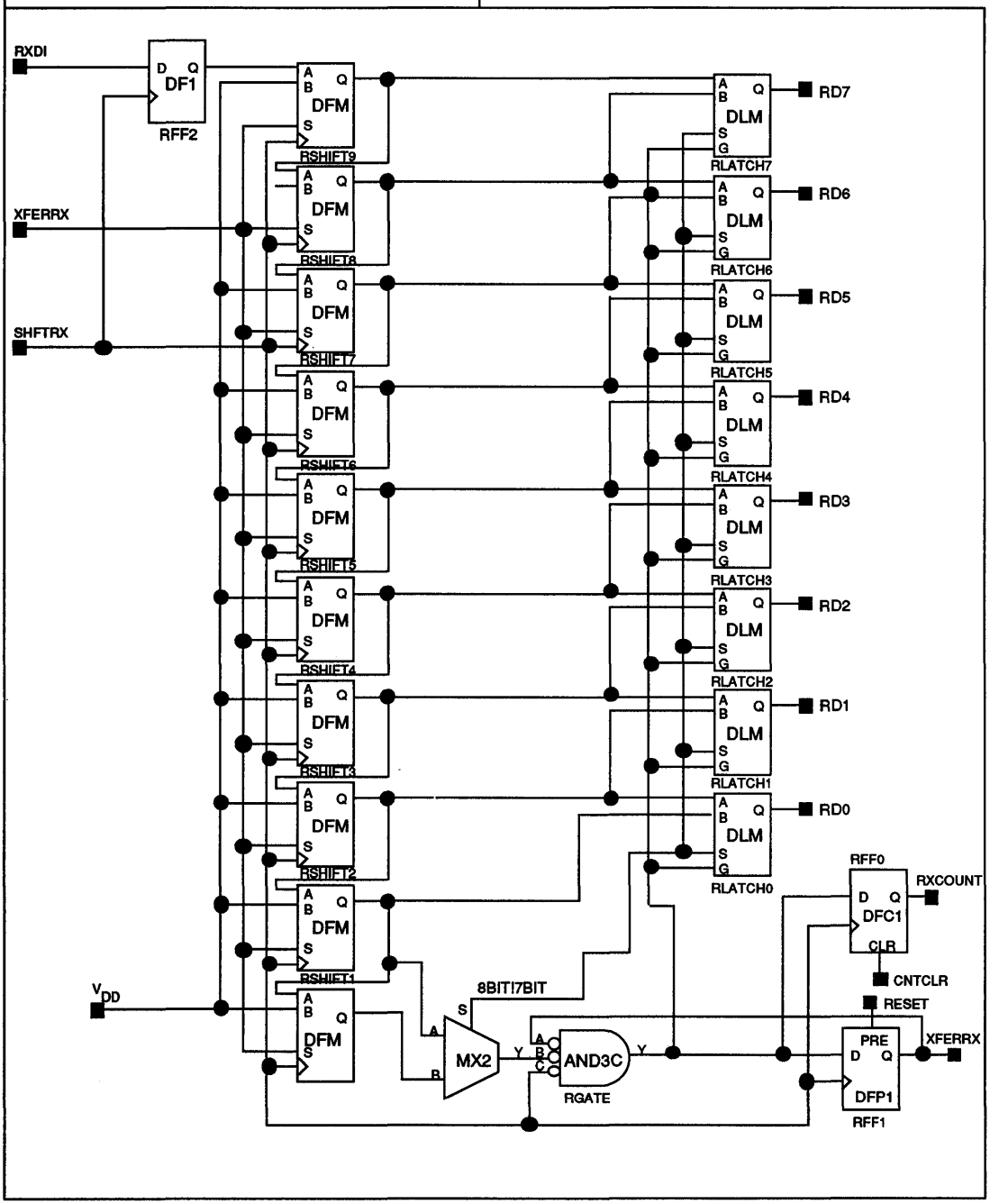
The receiver clock circuit is a 4-bit Gray code counter (similar to the transmitter clock circuit) and is driven by the same external clock signal (16BAUDCLK).

Figure 12: Internal Three-State Function Replaced with MUX



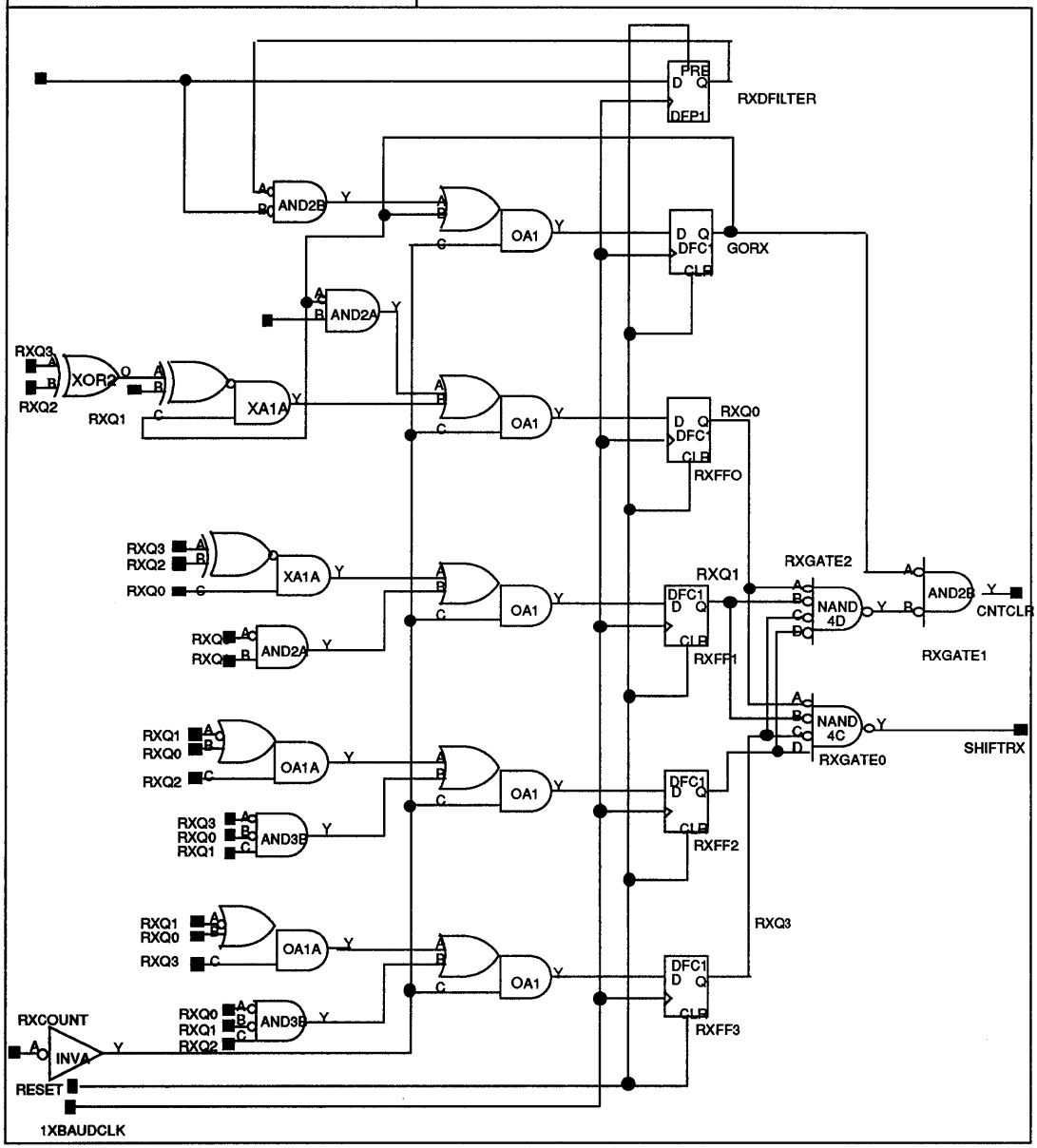
ACT 1 devices do not support internal three-state drivers; users implement internal bus designs using MUXs.

Figure 13: Receiver Shift Register and Buffer



The state machine has both synchronous and asynchronous resets (RXCOUNT, RESET). A filter circuit (RXDFILTER) prevents false starts due to errors on the data line (RXDI). The receiver clock (SHFTRX) is decoded from the outputs of the state machine RXQ[0:3].

Figure 14: Receiver Clock Circuit



Parity Checker

The receiver parity checker is programmable to either odd or even parity.

The parity checker generates a parity bit from the incoming serial data stream and compares it to the incoming parity bit. The parity error flag (PARITYERROR) is set accordingly. See Figure 15 below.

Design Complexity

The receiver section design required 72 logic modules.

Table 1 below provides a table of module count and percent utilization of the array.

Figure 15: Parity Error Generator

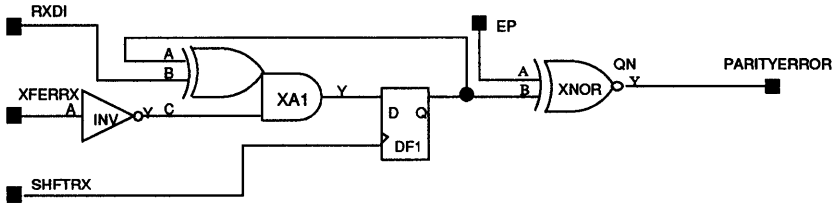


Table 1: Module Count and % Utilization

	Module Count
Transmitter	
Transmitter Buffer and Shift Register	40
Transmitter Clock Circuit	27
Transmitter Parity Generator	8
	Total 75
	25.4% Utilization of ACT 1010
Receiver	
Receiver Shift Register and Buffer	36
Receiver Clock	31
Receiver Parity Checker	5
	Total 72
	24% Utilization of ACT 1010

Applications Briefs:

Three-Stating ACT 1010/1020 Designs
ALU181
Fast Adders
8-bit Twos Complement Multiplier
The Actel Timer
Using the Actionprobes
Metastability

Three-stating ACT 1010/1020 Designs

During board test and debug it is frequently desirable to place all chip I/Os into a three-stated condition. This provides isolation from other three-stating circuit devices connected to signals common to the ACT device, as shown in Figure 1 below. The three-stated condition also allows board test for either trace integrity or insertion damage to ACT device pins.

Three-stating a design is easy using the unique debug features of ACT device architecture.

Three special pins on the ACT device facilitate three-stating:

Pin	68PLCC	84PLCC
MODE	54	66
SDI	56	72
DCLK	57	73

Pins SDI and DCLK are unassigned by the user or defined as input-only.

You may three-state all user-defined pins regardless of their normal mode definition: input-only output-only, or three-stateable. Each pin can be temporarily three-stated for test and debug purposes.

Figure 2 shows the sequence of three-stating. Seven data bits are clocked into the device, using the SDI pin as data input, and DCLK as clock. The mode pin distinguishes "test" mode from "normal." The data sequence clocked is {0001011}. After clocking the seventh bit, all user-defined pins enter a three-state condition until MODE is taken low.

Actel Actionprobes™ allow 100% observability of internal nodes; ACT devices may also be isolated from external board circuitry. Together, these two features provide a powerful debugging feature previously unavailable in custom devices.

Figure 1

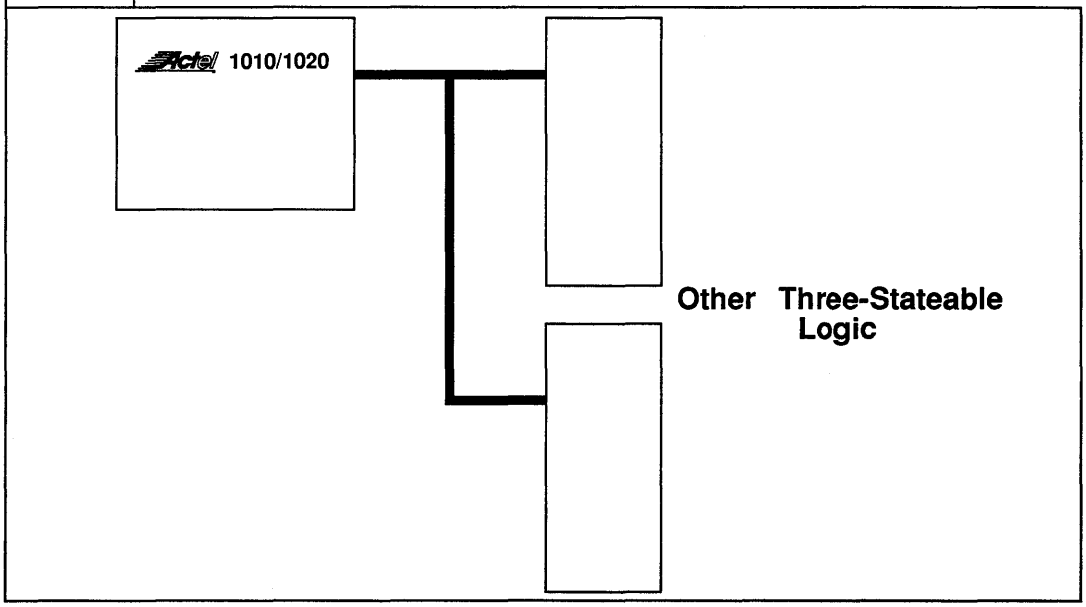
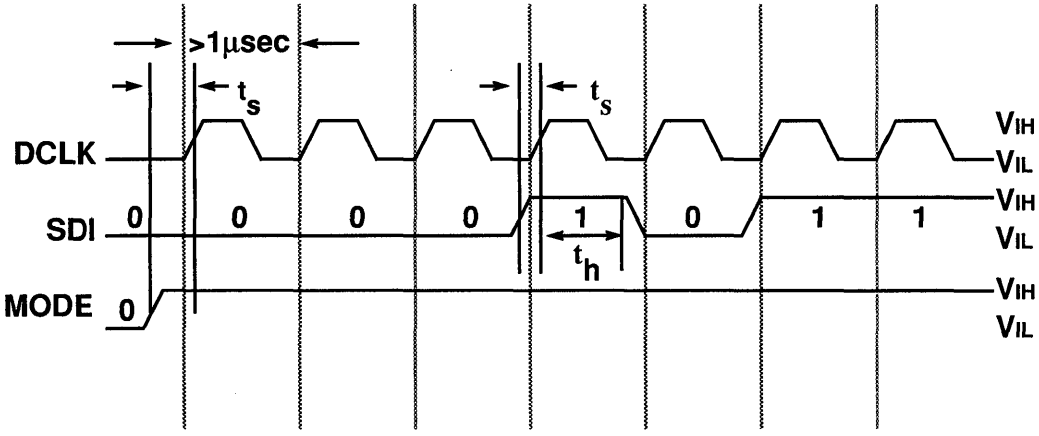


Figure 2



$\therefore$ $0v \leq V_{IL} \leq 0.5v$

$3.0v \leq V_{IH} \leq V_{CC}$

Notes:
Test mode configuration is a low frequency (<1.0Mhz) operation.
All setup and hold conditions (t_h, t_s)≥250 ns

Introduction

The venerable TTL 74181 is well known as it is useful. Its four select lines, mode and carry-in bits combine to offer over forty unique arithmetical and logical functions. Many designers dislike losing such functionality as they move from discreet to ASIC technology.

Designers may retain the function of the 74181 using Actel programmable gate arrays. Actel's macro library includes "T181," a soft macro functionally identical to the TTL component.

Implementation

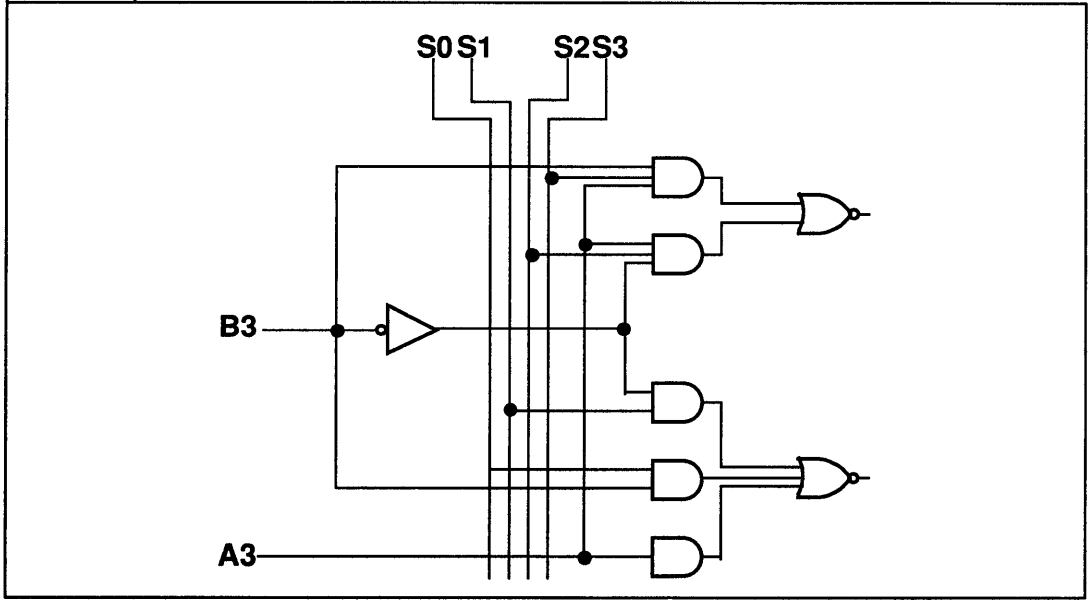
The 74181 depicted in popular TTL data manuals has six logic levels (seven for A=B output). While this representation may not be accurate at the transistor level, it typifies an ASIC design implementation.

The Actel design requires only four logic levels (five for A=B output). It was designed using components from Actel's library. The Actel logic module's flexibility enables such "logic compression" by implementing a wide variety of logical functions.

For example, each bit pair in the ALU passes through a section of logic, where select lines qualify the pair before outputting them to the remaining logic. The qualification logic, as it appears in TTL manuals, is shown in Figure. 1. It uses eight gates and three logic levels.

Figure 2 shows Actel's version of qualification logic. It required only *two* modules and *one* level of delay. The driving gate

Figure 1



TTL diagram. Bubbling (inverting) inputs on each gate driven by the section adjusts for this difference. The inversion is easily accomplished since all gates in the Actel library have versions with bubbled inputs.

One module implemented two three-input ANDs to two-input OR (AO32A), and the other two two-input ANDs to three-input OR (AO12A). Combined, these modules create the qualification logic.

The design's remaining logic is similar to that found in TTL manuals, with some improvements. M inputs required no inverter because M-driven gates had bubbled inputs. Single logic module macros implemented the ANDOR functions.

Another example of logic compression is in macro AOI32, used in the logic generating the F3 output. The macro implements the expression:

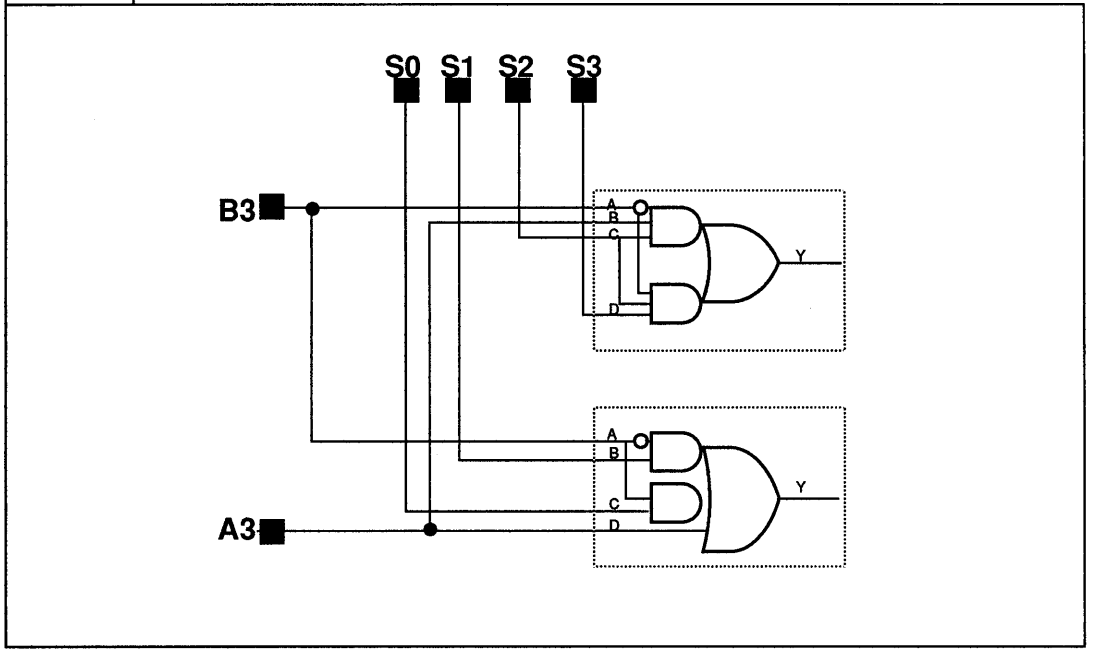
$$Y = /((/A*/B*/C) + (/A*/D))$$

in a single module. AOI32 clearly shows the savings in delay and modules achievable with the versatile Actel logic module.

Conclusion

Actel's ALU design uses only 32 logic modules, or about 10% of an ACT 1010 gate array. It is an excellent example of the Actel logic module's adaptability both in implementing logic functions, and in logic compression, for efficient, high-performance designs.

Figure 2



The Actel Family of Fast Adders

Adders, in a variety of sizes, are a very common function in digital designs. They are frequently implemented using carry-look-ahead (CLA) logic to speed carry propagation. However, as adder size increases, CLA logic requires increasingly wider gates and more logic levels, degrading adder performance.

Implementation

Actel solved the CLA logic problem by designing its family of fast adders using the Carry Select (CS) method. CS adders partition the addition into groups of bits. Two additions are performed simultaneously for all the groups. One addition is done with a carry-in of one and the other with a carry-in of zero. The two group sums and their carry-outs are connected to two input MUXs.

When determined, the carry from the lower groups selects the group which supplies the sum to the outputs. The carry-out for the group selects the sum and carry for the next higher group, and so on.

The speed of group additions and carry propagation determine the speed of the CS adder.

Group Size

Since carries ripple within groups, small group size is an advantage. Small groups, however, require more groups for a given number of bits, lengthening the carry propagation path. The design trade-off is thus the number of delays to a group sum versus the number of delays required to propagate a carry.

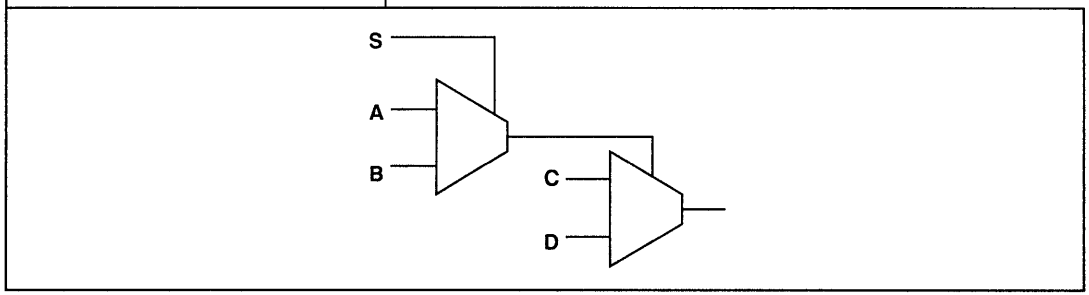
Optimizing the group sizes based on the adder delay makes Actel's adders are very fast.

Carry Propagation

As previously mentioned, carry propagation speed is a limiting factor in carry select adders performance. Group carries must be selected, just as the sums are. Unlike sums, carries select higher carries and so forth. This creates the problem of cascading carries through multiple logic levels.

Actel's logic module flexibility alleviates cascading carries by implementing a cascaded two-input MUX in a single module. The logic, shown in Figure 1, is a two-input MUX whose output drives the select for another two-input MUX. The macro allows two-level carry selection in one module and in only one delay. All Actel adder macros use the cascade MUX macro for rapid carry propagation.

Figure 1: Cascade MUX Macro



The Actel Adder Family

The following table lists Actel adders, their speeds (driving one load), and the number of logic modules used:

Macro Name	No. of Bits	Worst-Case Delay (ns)	Levels of Logic	No. of Modules
FADD8	8	36	4	37
FADD12	12	45	5	63
FADD16	16	47	5	78
FADD24	24	56	6	120
FADD32	32	65	7	160

Conclusion

Actel compresses Adder design logic using the Actel logic module to achieve a high level of performance.

Introduction

Multipliers are vital to systems design. In the past, designers integrated VLSI multiplier chips into their designs. As ASICs become more popular, ASIC-based systems increasingly utilize multiplier macros. Previously, no field programmable device had sufficient architectural flexibility and connectivity to efficiently implement multipliers. Actel has designed an eight-bit twos complement multiplier which easily fits on an ACT1010 or ACT1020 field programmable gate array (FPGA).

Algorithm

High-performance multipliers form the product of two numbers using combinatorial logic and an array of adders. Actel's multiplier is based on the Baugh-Wooley [1] algorithm (Equation 1) and uses conventional full adders for twos complement multiplication. When the size of the multiplier and multiplicand are known, the Baugh-Wooley formula can generate a series of product terms. In this example, when both multiplier and multiplicand are eight bits, M and N equal 8 in the formula.

Each product term contains "2" raised to a power. The product terms are arranged into an array according to their binary power. Adding the columns of the array results in the product of the two original operands.

In the generic architecture, AND gates generate the product terms, the adders sum them together. The adder array has a parallelogram shape. Dividing the parallelogram into four sections eases the design process. Each section generates a set of partial products, which are then added together

Equation 1

$$P = a_{m-1} b_{n-1} 2^{m+n-2} \sum_{i=0}^{m-2} \sum_{j=0}^{n-2} a_i b_j 2^{i+j}$$

$$+ 2^{m-1} \left(- 2^{n} + 2^{n-1} + a_{m-1} 2^{n-1} + a_{m-1} + \sum_{i=0}^{n-2} a_{m-1} b_j 2_i \right)$$

$$+ 2^{n-1} \left(- 2^{m} + 2^{m-1} + b_{m-1} 2^{m-1} + b_{m-1} + \sum_{i=0}^{m-2} a_i b_{n-1} 2^{i}\right)$$

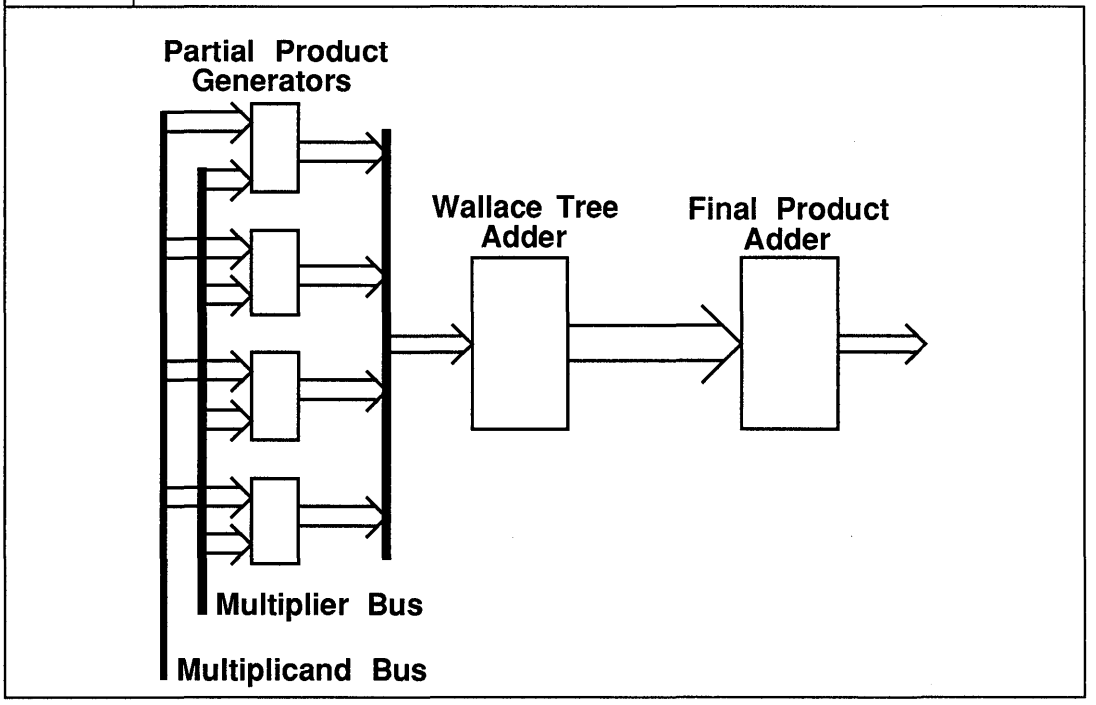
Actel Multiplier Architecture

The Actel multiplier macro architecture appears in Figure 1. It consists of three major sections: the partial product generators, Wallace tree adder, and final product adder.

The partial product generators implement the adder array sections using Actel library components. Since the partial product generators have similar structure, the design was copied and modified to create the others.

The Wallace tree adder sums the partial products and passes two binary numbers onto the final product adder. The final product adder performs arithmetic necessary to produce the product. The following is a discussion of each element in detail.

Figure 1



Partial Product Generator

Each partial product generator requires as input four bits from each operand. The bits pass through AND functions whose outputs pass through an array of adders to generate eight partial products. Because the partial product generators operate in parallel, all partial products are produced simultaneously.

The schematic has an array of adders with inputs either from an AND gate or the sum from another adder. The Actel library adder FA2A adds a two-input AND term with inverted inputs to another input and a carry-in. Most of the adders in the generator use FA2A, requiring only two logic modules to implement both the AND and full adder functions.

Wallace Tree

The multiplier's Wallace tree section sums columns of terms from partial product generators. Summing a column in binary arithmetic requires tracking carries generated to the higher power-of-two positions in the number. The Wallace tree uses three-bit adders to input up to three bits of the column. It feeds the adder sums into adders taking in other bits of the column, and feeds carry-outs to the inputs of adders summing the column in the next higher bit position, and so on. The tree output is a series of sum and carry-in bits which may be added in a conventional adder as two binary numbers, producing the final product.

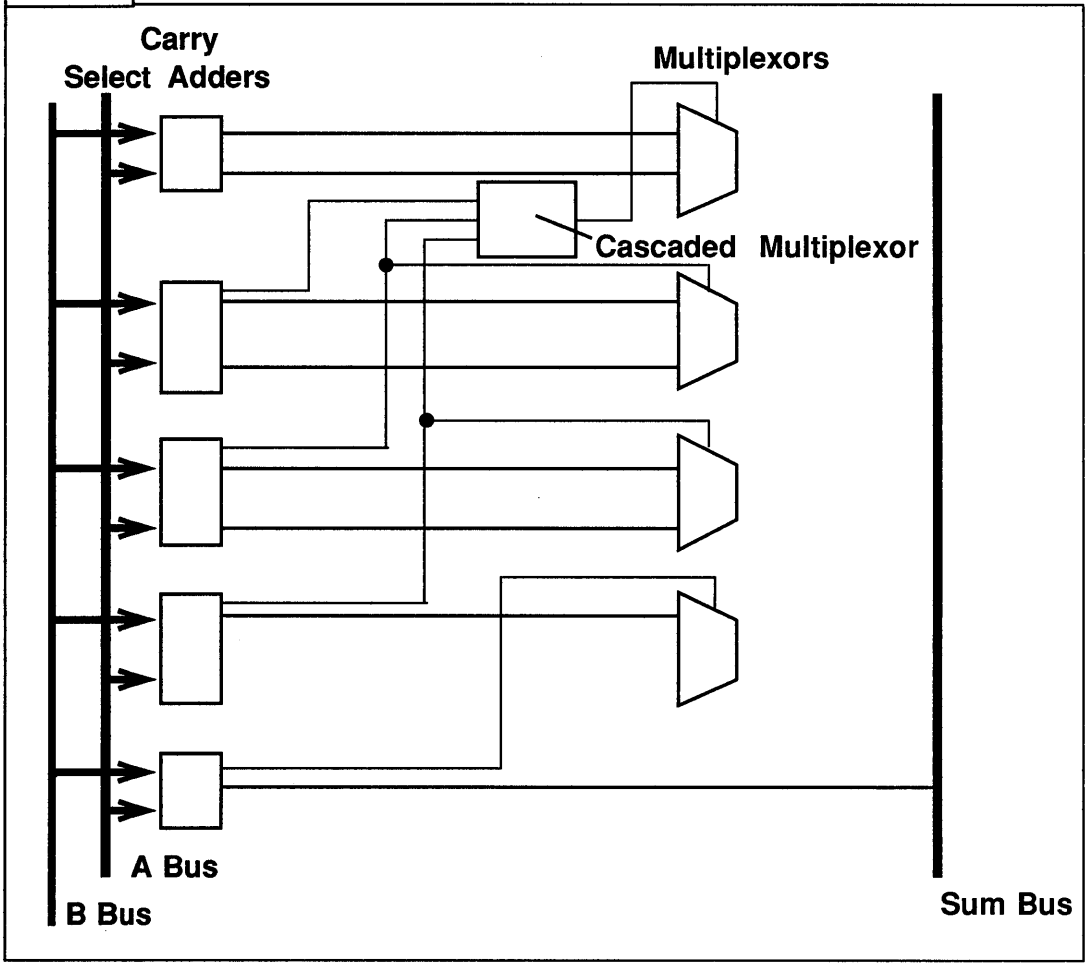
Because both carry-outs and carry-ins are active-low on the Actel adder macros, the adders can be used in a Wallace tree structure as a three-bit adder which outputs a sum and a carry. The carry-out from an adder in one column inputs directly to the carry-in of the next higher column. Other than the adders, no logic is required to design the Wallace tree.

Final Adder

The multiplier's last section is a binary adder. The Wallace Tree completely sums some lower columns of bits, which pass from the tree to the product bus. The tree outputs higher bits as two binary numbers to be added. The bits comprise the sum bit for each bit position and its respective carry-in. The final adder adds them together for the remaining bits of the product.

The design of the final adder uses the same design technique as the Actel family of fast adder macros. The architecture of the final product adder is shown in Figure 2 below.

Figure 2



The Carry Look Ahead Problem

Adders are usually implemented using carry look ahead (CLA) logic to speed carry propagation. However, as the size of the adder increases, the CLA logic requires wider gates and more logic levels, degrading adder performance.

Carry Select Addition

To solve the CLA problem Actel designs its fast adders using the Carry Select (CS) method. CS adders partition the addition into groups of two or three bits. Two additions occur simultaneously for all groups. One addition has a carry-in of one and the other a carry-in of zero. The two group sums and their carry-outs connect to two-input MUXs.

When determined, the carry from the lower groups selects the supply which sums the outputs. Next, the carry-out for the group is also selected and selects the sum and carry for the next higher group and so on.

The CS adder speed is determined by the speed of group additions and carry propagation speed.

Group Size

Since carries ripple within groups, small group size is an advantage. Small groups, however, require more groups for a given number of bits, lengthening the carry propagation path. The design trade-off is thus the number of delays to a group sum versus the number of delays required to propagate a carry.

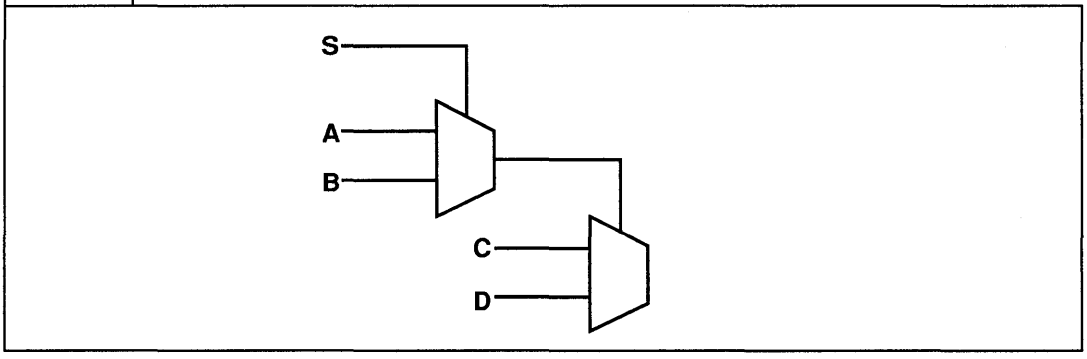
By optimizing group sizes based on adder macro delay parameters, Actel's adders are very fast.

Carry Propagation

As previously mentioned, carry propagation speed is a limiting factor in carry select adders performance. Group carries must be selected, just as the sums are. Unlike sums, carries select higher carries and so forth. This creates the problem of cascading carries through multiple logic levels.

Actel's logic module flexibility alleviates cascading carries by implementing a cascaded two-input MUX in a single module. The logic, shown in Figure 3, is a two-input MUX whose output drives the select for another two-input MUX. The macro allows two-level carry selection in one module and in only one delay. All Actel adder macros use the cascade MUX macro for rapid carry propagation.

Figure 3



The Actel Advantage

The eight bit twos complement multiplier illustrates the types of macros implemented on Actel programmable gate arrays. The design method detailed here suits multipliers of many different configurations. This technique also applies to 3X7, 9X4, or 6X5 multipliers depending on system requirements.

Other multiplier algorithms also apply. A magnitude array multiplier design using the Cavanagh method [1] requires less logic than the eight bit multiplier above. Other alternatives include the Booth and Bit-Pair Recording methods [1].

The multiplier macro may be designed as part of the logic in a field programmable gate array. The remaining gate array modules are then available for other design purposes, either other arithmetic functions or glue logic. For example, Actel's 8X8 multiplier macro uses 241 of 546 modules on an Actel 1020. Nearly 60% of the chip is still available for other purposes.

In any design, Actel's provides the technology to complete the entire gate array development process including schematic capture, simulation, place and route, timing analysis, and fusing the chip.

1. Cavanagh, Digital Computer Arithmetic Design and Implementation, McGraw-Hill, New York. 1984.

Introduction

The Actel Timer is an interactive, menu-driven tool for analyzing design timing. It automatically provides delay data for all paths in a design between clocked macros, and for any other paths specified by the user.

You may review the Timer output as total path delay or break up the path to show the delay through each module.

Functional Description

When first entering the Timer, it prompts you to select operating conditions for your device: typical, commercial or industrial. The temperature and voltage range of each condition is shown in Table 1. You may also choose to inspect either Post-layout or pre-layout calculations. Pre-layout delays are estimates based on statistical propagation delays and loading. Post-layout delay information includes placement and routing data specific to the design, and thus is the most precise timing specification.

Pin Sets

The Timer automatically extracts all delays between the pins of clocked macros and places the delay information in a default pin set. This information is useful for examining the delays between the pins, *and* it provides a quick reference for the design's maximum clock period. In addition, the information may be easily manipulated to reveal clock line skew, or any other signal skew in the design.

It is easy to create any number of beginning or ending pin sets, of any size, and extract the delays between them. These pin sets can be saved and recalled to quickly observe the paths between any two sets

Pins may be added to or removed from any pin set. Timer commands also may logically "AND" or "OR" groups of pins to create new groups of pins.

Table 1: Operating Conditions

Delay	Temp(°C)	Voltage Range(V)
Typical	25	5.0
WC Com	0-70	4.75-5.25
WC Ind	-40 - 85	4.50-5.50

Display

The Timer displays delays from module input pins to module input pins. For example, "delay1" in Figure 1 below includes the delay of gate "**G0**" plus the net delay of "**Net0.**" The Timer provides a variety of options for design inspection and analysis.

The "Pins" and "More" command allow you to look at all the delays between the pin sets. "Pins" shows the ten longest delays in descending order. "More" shows the next ten slowest paths, and, if repeated, successively displays all remaining delays in the set. The "Pins" command display appears in Figure 2. The information includes the delay rank, the delay in nanoseconds, and specifies the path end points.

Figure 1

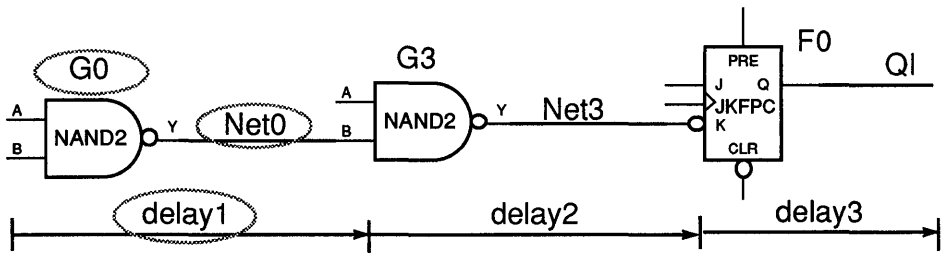


Figure 2: "Pins"

1st longest path to all endpin

Rank	Total	Startpin	First Net	End Net	End Pin
0	84.2	F4:CLK	NET1	NET2	F1:J
1	83.4	F17:CLK	NET3	NET4	F8:J
			•		
			•		
			•		
8	77.0	F5:CLK	NET30	NET21	F0/U0:K
9	76.1	F6:CLK	NET32	NET32	F0/U0:J

The "Path" command shows the delay through any specified path in more detail. As shown in figure 3, "Path" breaks down the delay through a path to display it on a module-by-module basis, showing exactly how each module contributes to the overall delay of the path. The information displayed includes the cumulative delay, the delay through the macro, delay type, number of loads, macro name, and net names.

The "Path" command helps fine tune design timing. For example, if a path delay is too long, it shows if a heavily loaded macro is slowing the path. You may reduce that load using parallel logic to improve the performance of the path.

Figure 3: "Path"

1st longest path to U0/U7/F0/U0:K (falling)

Total	Delay	Src	Load	Macro	Starting pin	Net name
82.1	6.5	Tsu	0	JKF_0	G12:B	NET X
75.6	7.0	Tpd	1	NAND2	G3:A	NET Y
68.8	11.8	Tpd	2	XOR	G1:A	NETA
56.8	10.8	Tpd	1	NOR2A	G14:A	NETB
46.0	10.0	Tpd	1	NOR4A	G13:C	NETC
36.0	16.4	Tpd	7	AX1	G5:C	NETD
19.6	7.1	Tpd	1	AND2A	G4:A	NETE
12.5	12.5	Tco	6	JKF_1	U1:CLK	NETF
0.0	0.0	Skw	16		U0:CLK	NETG

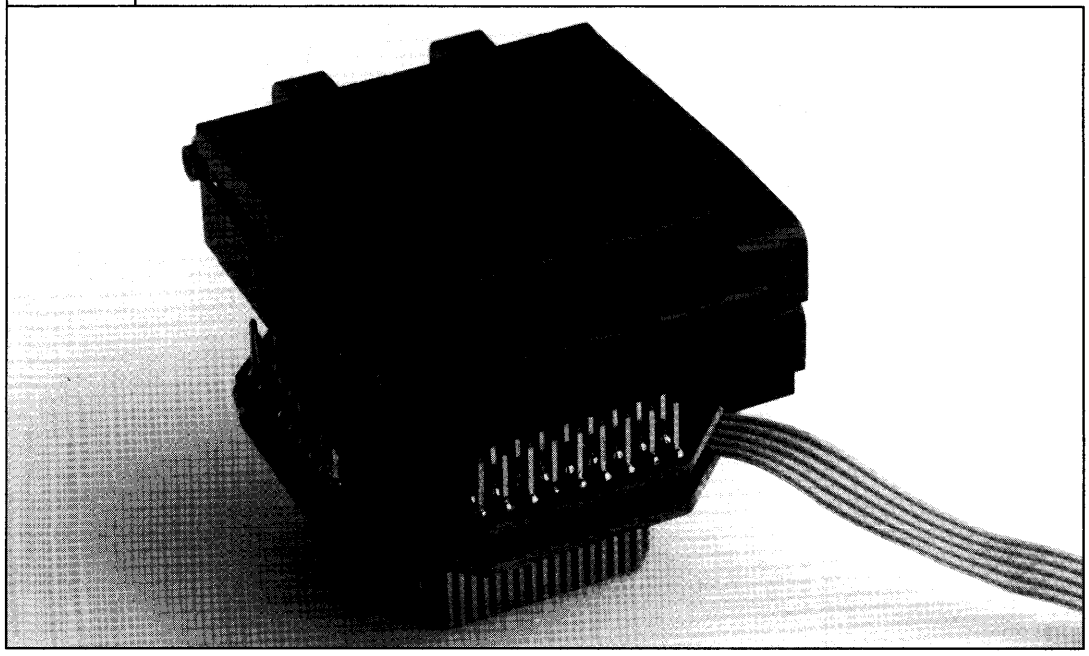
Introduction

Actel's Actionprobe™ circuitry permits external monitoring of ACT1 device internal signals *after* device configuration. This unique diagnostic feature allows 100 percent observability of the circuit. Observability reduces the time required for design verification and test vector generation; it also facilitates system troubleshooting. ACT1010 and ACT1020 devices each have two dedicated probe pins, PRA and PRB. After device configuration, the Debugger software and Actionprobe hardware permits the connection of any two signal nodes in the device to the Actionprobe pins. Signal nodes may be freely changed under software control to any nodes in the design.

Set Up

Actionprobe hardware consists of a tower probe with a footprint of a 68-pin or 84-pin LCC package. A ZIP socket on top of the tower probe holds the programmed ACT device in place (Figure 1). The Activator bus board on the PC drives the control signals SDI, DCLK, MODE and GND through a connector to the chip. SDI receives the serial addresses of the internal nodes from the Activator bus board. DCLK clocks the serial address into the device. The MODE pin determines whether the device is in debug mode. For device debugging on the user's circuit board, SDI, DCLK and MODE are terminated to ground through a >10K Ohm resistor. Actionprobe pins may connect directly to a logic analyzer or oscilloscope.

Figure 1



With a programmed ACT device in the Actionprobe socket, you can plug it directly into the system board. The device receives real time stimuli from the target board at the same time it is verified and debugged. See Figure 2 below.

In Circuit Probing

To move to other internal nodes, you simply change the node names. The new node addresses are automatically shifted in the device. The new nodes are brought out to the Actionprobe pins.

The "ICP" (In-Circuit-Probing) command connects the Actionprobe pins to internal node. The syntax is :

ICP node_1 node_2

where node name "node_1" is electrically connected to PRA and node name "node_2" goes to PRB.

Internal nodes up to 15 MHz can be externally monitored. Internal signals pass through an inverting buffer prior to the Actionprobe pin.

One hundred percent observability of all internal device signals is a feature unique to Actel, unavailable in conventional masked gate-arrays or programmable logic devices.

Actionprobe Calibration

Since the Actionprobe circuitry does not introduce any additional loading to the design, the AC characteristic of the observed internal nodes remains unchanged. Also, because the Actionprobe propagation delay is independent of layout, the Actionprobe delay remains unchanged for all points in the device.

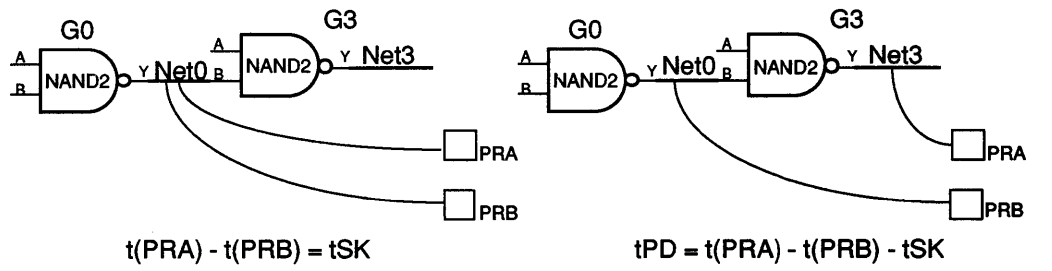
Figure 2

With the Actionprobes, you may calibrate for accurate measurement of propagation delays. This is done by connecting both Probe PRB pins to the same point in the device to determine the delay difference and subtracting that difference in subsequent measurements. For example, the user can electrically connect both Actionprobe pins to node Net0 in Figure 3. The delay difference is the skew of the Actionprobe pins. In the Debugger, the user can move the slower Actionprobe pin to node Net3. The propagation delay from the output of G0 to the output of G3 is the showing delay difference between the Actionprobe pin less the Actionprobe skew time.

Pin Assignment for the Dedicated Pins (SDI, DCLK, PRA, PRB).

During device verification, the dedicated pins SDI, DCLK and the Actionprobe pins PRA, PRB are used as special I/O pins for debugging purpose. These pins are assigned as non-critical so the design is functional without them. After device verification, you can permanently assign these pins (SDI, DCLK, PRA, PRB) as regular I/Os by programming the security fuses. The Actionprobe pins are then disabled to prevent unauthorized device probing.

Figure 3



Actel Metastability Characteristics

Designers often have asynchronous signals coming into synchronous systems. Typically a flip-flop is used to synchronize the incoming signal with the system clock.

If the asynchronous incoming signal does not meet the setup time requirement for the flip-flop, there exists a window of time where the incoming signal may cause the flip-flop to develop an unknown or metastable condition. Figure 1 shows the time window (Tw) where the DATA IN signal causes metastability. How long a flip-flop will remain in a metastable condition before its output resolves to either logic level is probabilistic and can be expressed as follows:

$$MTBF = \frac{1}{Fclk * Fdat * C1\ e^{-C2 \bullet tr}}$$

where the constants depend on the ACT 1 device characteristics, and:

Fclk = System clock frequency (MHz);

Fdat = Incoming data rate (MHz);

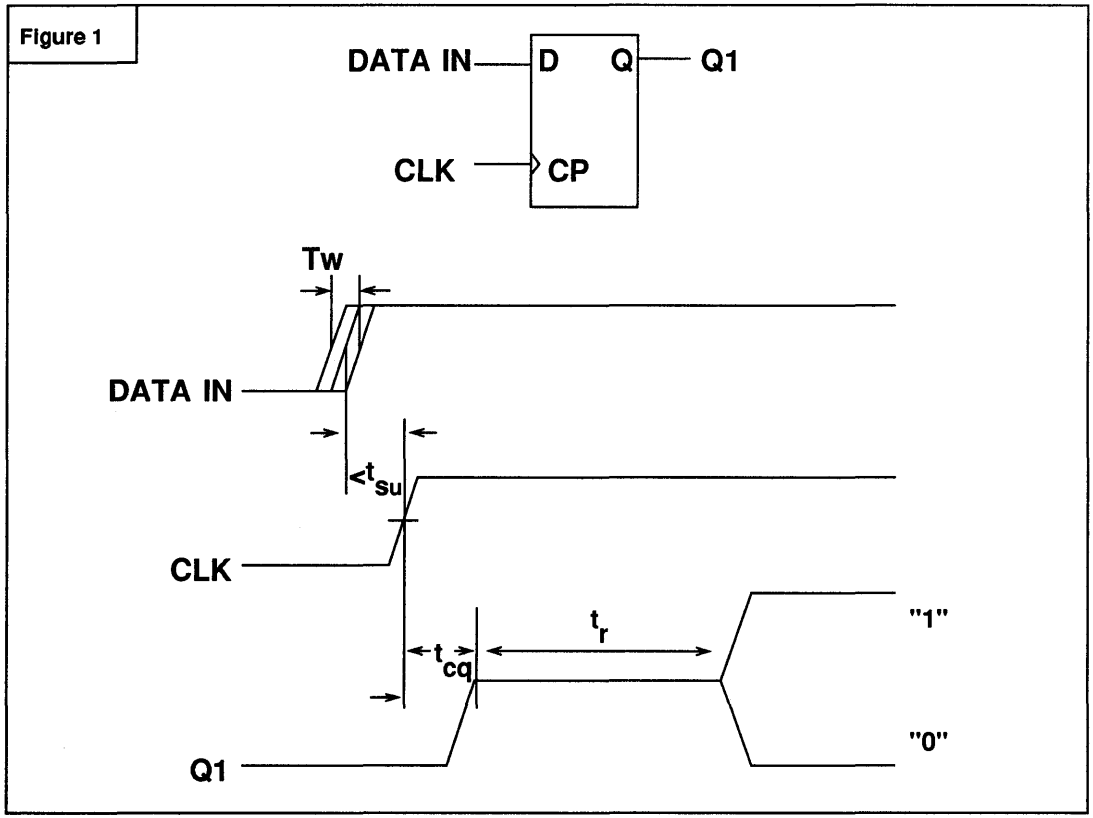
e = Natural log base

tr = Resolution time

C1 = 10^3 (/MHz);

C2 = 4.6052 (/ns)

Figure 1



With the incoming data rate and system clock frequency you can calculate how often a metastable state of a given duration will occur. To design your system to accommodate acceptable occurrence of metastability, either double clock the incoming signal or allow additional delays between flip-flop outputs to the next clocked device to allow for metastability resolution.

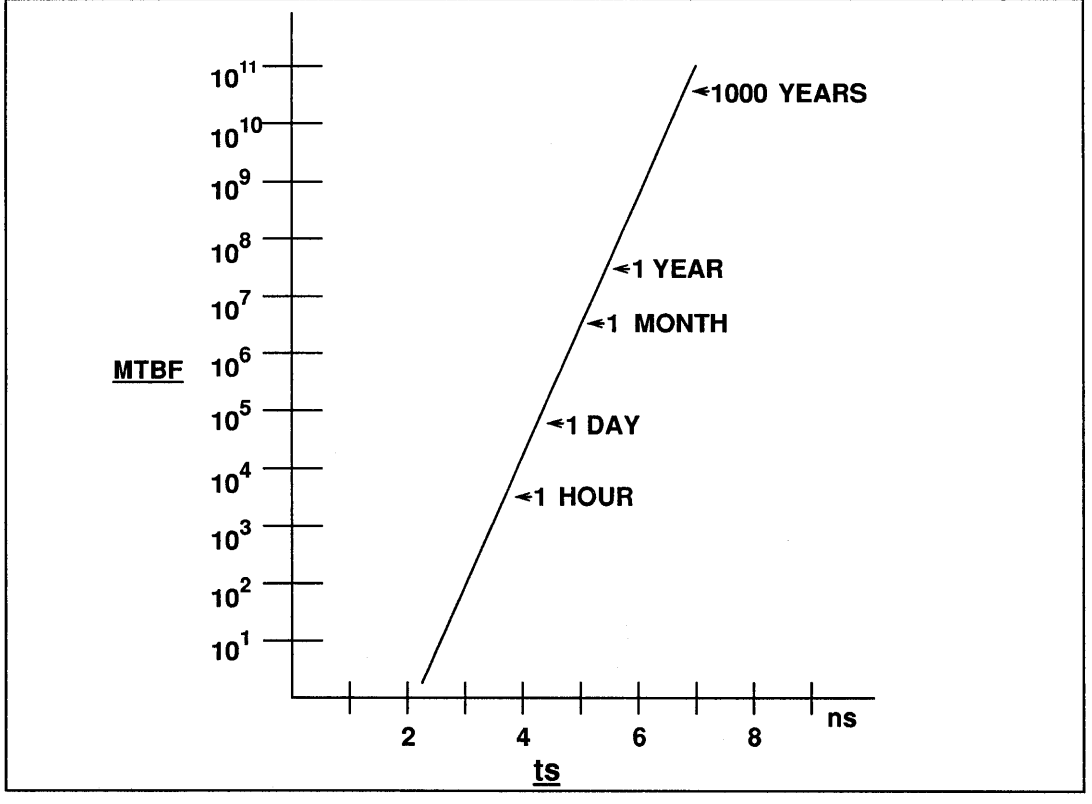
Sample Calculation

For a system with a clock of 10 MHz and a data rate of 1MHz different metastability resolution times yield the following MTBF:

tr	MTBF
3.0 nS	10^2 Sec.
4.0 nS	10^4 Sec.
6.0 nS	10^8 Sec.

Plot 1 shows a graph of the metastability duration versus frequency of occurrence for the example system. As the plot shows, linear increases in resolution time cause exponential increases in MTBF.

Plot 1: Metastable MTFB as a Function of Resolution Time



ACTEL CORPORATION
DIRECT SALES OFFICES

ACTEL CORPORATION
955 E. Arques Ave.
Sunnyvale, CA 94086
Tel: (408) 739-1010

ACTEL CORPORATION
22750 Hawthorne Blvd., Suite 215
Torrance, CA 90505
Tel: (213) 378-0656

ACTEL CORPORATION
1740 Mass Avenue
Boxborough, MA 01719
Tel: (508) 635-0010

ACTEL CORPORATION
2350 Lakeside Blvd., Suite 850
Richardson, TX 75082
Tel: (214) 235-8944

DOMESTIC REPRESENTATIVES

ALABAMA
Rep Inc. (205) 881-9270
ARIZONA
Luscombe Engineering (602) 949-9333
CALIFORNIA
Centaur (Calabasas) (818) 704-1655
Centaur (Irvine) (714) 261-2123
Centaur (San Diego) (619) 278-4950
I² Inc. (Santa Clara) (408) 988-3400
I² Inc. (Orangevale) (916) 989-0843
COLORADO
Luscombe Engineering (303) 772-3342
CONNECTICUT
CompRep Associates (203) 269-1145
FLORIDA
Sales Engineering Concepts (Altamonte Springs) (407) 682-4800
Sales Engineering Concepts (Deerfield Beach) (305) 426-4601
GEORGIA
Rep Inc. (404) 938-4358
ILLINOIS
Carlson Electronic Sales Associates (312) 956-8240
INDIANA
SAI Marketing Corp. (317) 545-1010
IOWA
Carlson Electronic Sales Associates (319) 378-1450
KANSAS
DLE Electronics (316) 683-6400
MARYLAND
New Era Sales (301) 544-4100
MASSACHUSETTS
CompRep Associates (617) 329-3454
MICHIGAN
SAI Marketing Corp. (313) 750-1922
MINNESOTA
Mel Foster Technical Sales (612) 941-9790
MISSOURI
John Macke Co. (314) 432-2830
NEW JERSEY
Nexus (201) 947-0151
NEW YORK
L-MAR Associates (Apalachin) (607) 687-1828
L-MAR Associates (E. Rochester) (716) 381-9100
L-MAR Associates (Poughkeepsie) (914) 462-8025
NORTH CAROLINA
Rep Inc. (Charlotte) (704) 563-5554
Rep Inc. (Morrisville) (919) 469-9997
OHIO
SAI Marketing Corp. (Columbus) (614) 451-0778
SAI Marketing Corp. (Dayton) (513) 435-3181
SAI Marketing Corp. (Warrensville Heights) (216) 591-0530
OREGON
L² LTD. (503) 629-8555
PENNSYLVANIA
Omega Sales (215) 947-4135
TENNESSEE
Rep Inc. (615) 475-4105
TEXAS
OM Associates (Austin) (512) 388-1151
OM Associates (Houston) (713) 789-4426
OM Associates (Richardson) (214) 690-6746
UTAH
Luscombe Engineering (801) 565-9885
WASHINGTON
L² LTD. (206) 827-8555
WISCONSIN
Carlson Electronic Sales Associates (414) 476-2790
CANADA
Clark-Hurman Associates (Quebec) (514) 426-0453
Clark-Hurman Associates (Ontario—Brampton) (416) 453-1118
Clark-Hurman Associates (Ontario—Nepean) (613) 727-5626

DOMESTIC DISTRIBUTORS

WYLE LABORATORIES

ARIZONA
Phoenix (602) 437-2088
CALIFORNIA
Calabasas (818) 880-9000
El Segundo (213) 322-8100
Irvine (714) 863-9953
Rancho Cordova (916) 638-5282
San Diego (619) 565-9171
Santa Clara (408) 727-2500
COLORADO
Thornton (303) 457-9953
OREGON
Hillsboro (503) 640-6000
TEXAS
Austin (512) 834-9957
Houston (713) 879-9953
Richardson (214) 235-9953
UTAH
West Valley City (801) 974-9953
WASHINGTON
Redmond (206) 881-1150

PIONEER STANDARD ELECTRONICS

CONNECTICUT
Norwalk (203) 853-1515
ILLINOIS
Addison (312) 495-9680
INDIANA
Indianapolis (317) 573-0880
KANSAS
Lenexa (913) 492-0500
MASSACHUSETTS
Lexington (617) 861-9200
MICHIGAN
Grand Rapids (616) 698-1800
Livonia (313) 525-1800
MINNESOTA
Eden Prairie (612) 944-3355
MISSOURI
St. Louis (314) 432-4350
NEW JERSEY
Pine Brook (201) 575-3510
NEW YORK
Binghamton (607) 722-9300
Fairport (716) 381-7070
Woodbury (516) 921-8700
OHIO
Cleveland (216) 587-3600
Dayton (513) 236-9900
Reynoldsburg (614) 221-0043
PENNSYLVANIA
Pittsburg (412) 782-2300

PIONEER TECHNOLOGIES

ALABAMA
Huntsville (205) 837-9300
FLORIDA
Altamonte Springs (407) 834-9090
Clearwater (813) 536-0445
Deerfield Beach (305) 428-8877
GEORGIA
Norcross (404) 448-1711
MARYLAND
Gaithersburg (301) 921-0660
NORTH CAROLINA
Charlotte (704) 527-8188
Durham (919) 544-5400
PENNSYLVANIA
Horsham (215) 674-4000
WASHINGTON, D.C. (301) 921-0660

INTERNATIONAL DISTRIBUTORS

DENMARK
Nordisk Electronik AS (Herlev) (2) 84.20.00
ENGLAND
Gothic Crellon Limited (Wokingham) (734) 78.88.78
Manhattan Skyline, Ltd. (Maidenhead) (628) 75.85.1
FINLAND
OY Fintronic AB (Helsinki) (0) 69.26.022
FRANCE
ASAP (Montigny Le Bretonneux) (1) 30.43.8233
Scaib S.A. (Rungis Cedex) (1) 46.87.2313
NORWAY
Nordisk Electronik AS (Hvalstad) (2) 84.50.70
SWEDEN
Traco AB (Farsta) (8) 93.00.00
SWITZERLAND
Omni Ray AG (Dietikon) (1) 835.21.11
WEST GERMANY
bit-electronic AG (Munich) (89) 41.80.070
ECN/DACOM (Munich) (89) 96.09.080

Actel Corporation
955 East Arques Avenue
Sunnyvale, CA 94086
408.739.1010

5172011-0/10K