I-Cube®

Digital Crosspoint Switching CD Contents

I-Cube®

Product Selection Guide

IQ BitSwitch Products

Product	Speed	TQ52	PQ80	J84	TQ100	PQ144	TQ144
IQ96	-10					X	X
IQ64B	-10			X	X		
IQ48	-7		X				
IQ32B	-7	X					

IQX Enhanced BitSwitch Products

Product	Speed	PQFP84	PQ208	PQ304	PB416
IQX320	-12				X
	-10				X
IQX240B	-12			X	
	-10			X	
IQX160	-10		X		
	-7		X		
IQX128B	-10	X			
	-7	X			

PSX BusSwitch Products

Product	Speed	PQ160	PQ208	PQ240
PSX160	-100			X
	-133			X
PSX128B	-100		X	
	-133		X	
PSX96B	-100	X		
	-133	X		

Package(s)
J=PLCC, PB=PBGA, PQ=PQFP, TQ=TQFP

Temperature
Commercial temperature 0 to 70 degrees

DESIGN TOOLS

Product	**Platform**	**Device(s) Supported**
Development Software		
IDS100	PC	IQ, IQX, PSX
IDS200	Sun	IQ, IQX, PSX
Starter Kits		
IQBIQX320SK	PC	IQX320
IDBIQX160SK	PC	IQX160
IDB96SK	PC	IQ96
IDBPSX160SK	PC	PSX160

LITERATURE

Technical Seminar

Digital Crosspoint Switching	Technical Seminar Presentation	*Presentation with notes describing products, design tools and application examples*

Datasheets

Digital Crosspoint Switch Products	IQ Family Datasheet	*IQ96 - IQ32B devices*
	IQX Family Datasheet	*IQX320 - IQX128B devices*
	PSX Family Datasheet	*PSX160 - PSX96B devices*
Design Tools	IDS100 Datasheet	*PC version of design software*
	IDS200 Datasheet	*Sun SparcStation version of design software*
	Starter Kit Datasheet	*IQX, IQ, PSX starter kits*

Reference Designs

Complete system designs with VHDL code

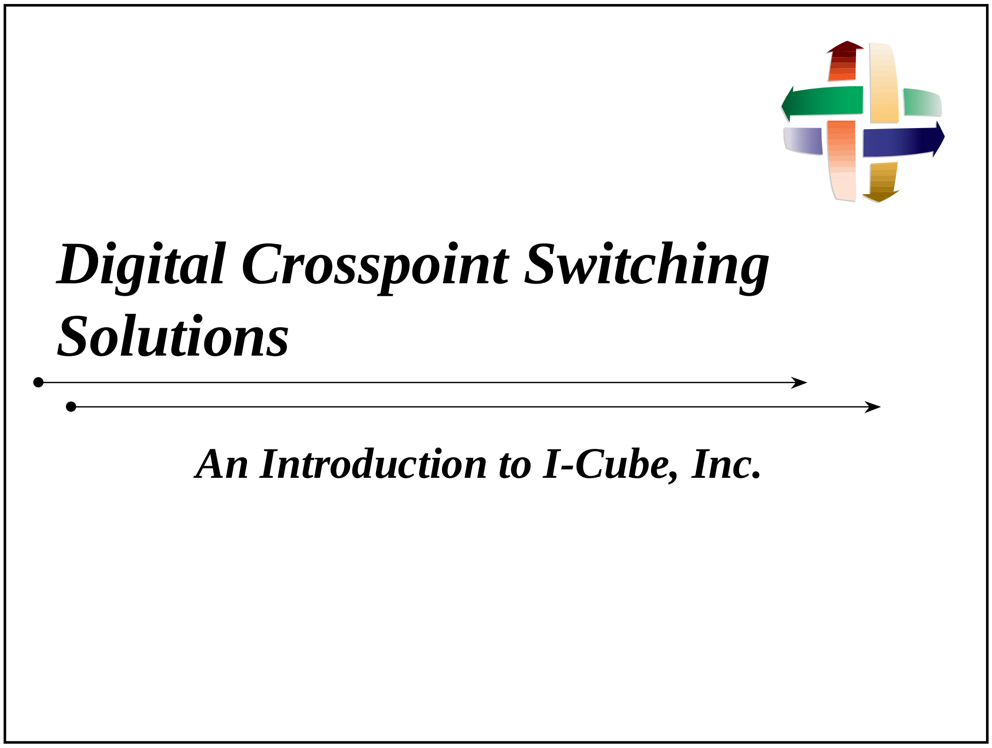
Application Briefs	Optimal I/O Assignment in an IQX-based Crossbar Switch	*IQX Family*
	Replacing a LSI Logic L64270 with an I-Cube PSID	*64 x 64 switch using IQX160/128B : Includes VHDL code*
Application Notes	Configuring I-Cube's PSID Devices	*IQ, IQX, PSX*
	JTAG UART for Controlling PSIDs	*IQ, IQX, PSX : Includes VHDL code*
	Designing a Large Crossbar with Fast Reconfiguration	*256 x 256 switch using 4 x IQX320 : Includes VHDL code*
	RapidConnect Controller for PSX Devices	*PSX160/128B/96B*
	Designing a 128x128 Cross With Fast Reconfiguration	*128 x 128 switch using IQX320 : Includes VHDL code*
	Implementing Bank Switching Using IQX Devices	*IQX160 : Includes VHDL code*
	An 8 input x 8 output Cell Switch Using PSX Devices	*8 x 8 x 32-bit bus switch using 4 x PSX128B : Includes VHDL code*

Software Documentation

IDS100 Users Manual(s)	IQX Family Users's Guide	*How to generate bitstreams using IDS100 for IQX devices*
	IQ Family Users's Guide	*How to generate bitstreams using IDS100 for IQ devices*
	PSX Family Users's Guide	*How to generate bitstreams using IDS100 for PSX devices*
IDS200 Users Manual	IQX Family Users's Guide	*How to generate bitstreams using IDS200 for IQX devices*
	IQ Family Users's Guide	*How to generate bitstreams using IDS200 for IQX devices*
	PSX Family Users's Guide	*How to generate bitstreams using IDS200 for IQX devices*
Register Programming Manuals	IQX Family Register Programming User's Reference	*Register level details for IQX devices for embedded bitstream generation*
	IQ Family Register Programming User's Reference	*Register level details for IQ devices for embedded bitstream generation*
	PSX Family Register Programming User's Reference	*Register level details for PSX devices for embedded bitstream generation*

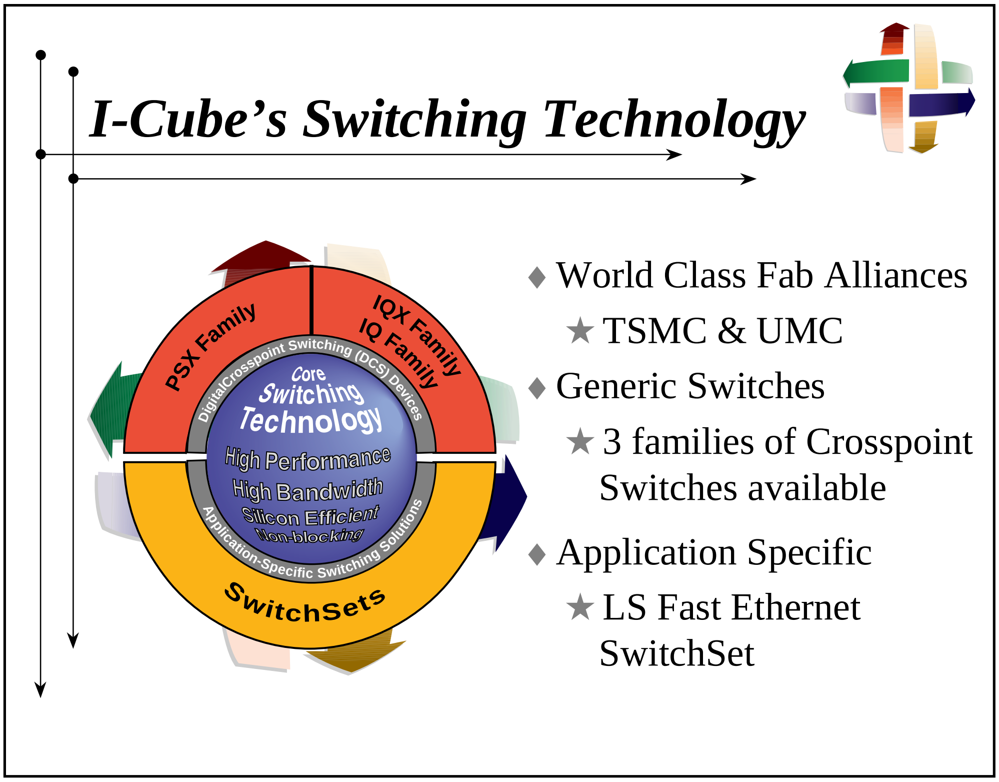
Introduction to Crosspoint Switching

- I-CUBE OVERVIEW
- DIGITAL CROSSPOINT SWITCH PRODUCT OVERVIEW
- DEVELOPMENT TOOLS
- TARGET APPLICATIONS & SOME DESIGN EXAMPLES



Digital Crosspoint Switching Solutions

An Introduction to I-Cube, Inc.



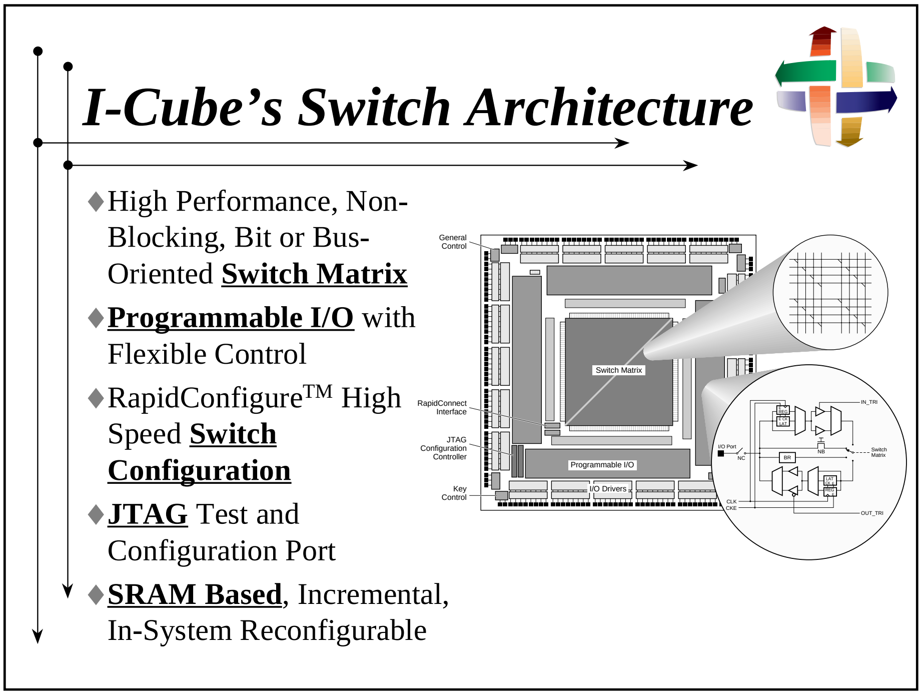
I-Cube’s Switching Technology

- ♦ World Class Fab Alliances
  - ★ TSMC & UMC
- ♦ Generic Switches
  - ★ 3 families of Crosspoint Switches available
- ♦ Application Specific
  - ★ LS Fast Ethernet SwitchSet

I-Cube’s mission is to develop semiconductor solutions for digital switching applications based on the company's proprietary, high performance switching technology. The heart of I-Cube’s products is the non-blocking, high performance silicon switching technology, which is heavily patented. The company has leveraged its core switching technology with both it’s generic Crosspoint Switches, also known as Digital Crosspoint Switching (DCS), and application specific chip sets (SwitchSets™).

I-Cube is a privately held company headquartered in Campbell, CA. I-Cube’s products are designed in-house and manufactured using the leading edge mainstream process technology offered by selected foundry partners. Devices are assembled and tested using multiple domestic and off-shore sources. I-Cube uses a network of domestic distributor and sales reps, and international stocking reps.

DCS devices are used in a wide range of applications including: telecommunications, networking, and digital video/imaging applications. SwitchSets are targeted at LAN/WAN and Gigabit switching.



I-Cube's Switch Architecture

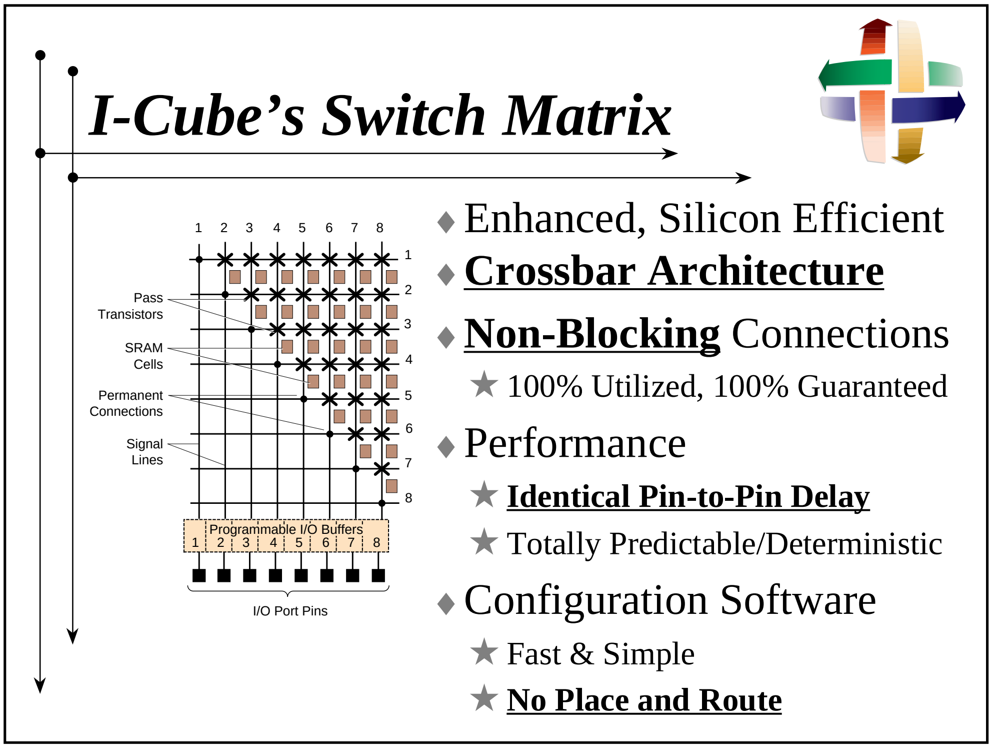
- High Performance, Non-Blocking, Bit or Bus-Oriented **Switch Matrix**
- **Programmable I/O** with Flexible Control
- RapidConfigure^{TM} High Speed **Switch Configuration**
- **JTAG** Test and Configuration Port
- **SRAM Based**, Incremental, In-System Reconfigurable

The architecture of the crosspoint switching devices consists of four basic blocks;

1. The switch matrix
2. The programmable I/O blocks
3. The configuration and switch control
4. The clocking and tri-state controls (port control signals)

Each programmable I/O port on the switching device is connected to a unique line in the switch matrix. Digital signals are brought into the device through I/O ports that are configured as inputs, the signals are then switched using the switch matrix, and they go out over the I/O ports that are configured as outputs. These devices use SRAM based configuration and therefore must be configured in-system every time the power is turned on.

One of the key advantages offered by I-Cube's switching devices is the flexibility they offer the system designer. The flexibility manifests itself in multiple ways; the I/Os on the device are identical and user programmable. You can make them into input, output or bidirectional. You can make 1-to-1 (point-to-point), 1-to-many (point-to-multi point) or many-to-one connections through the switching core. The data flowing through the device can be flow through, registered or latched - this in combination with programmable clocks and clock polarity can be used very effectively in high performance designs where the margins on set up and hold times are very tight. The device can be configured incrementally so you can alter I/O port & connection configurations without affecting data integrity on any "live" paths in the device.



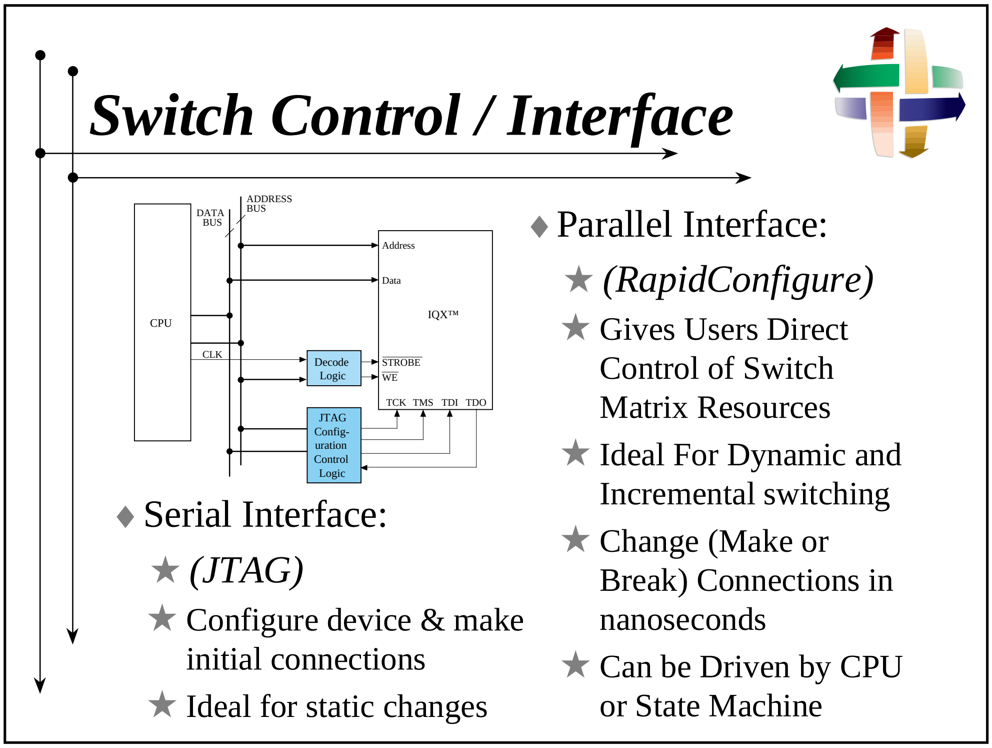
I-Cube's Switch Matrix

- ♦ Enhanced, Silicon Efficient
- ♦ **Crossbar Architecture**
- ♦ **Non-Blocking** Connections
  - ★ 100% Utilized, 100% Guaranteed
- ♦ Performance
  - ★ **Identical Pin-to-Pin Delay**
  - ★ Totally Predictable/Deterministic
- ♦ Configuration Software
  - ★ Fast & Simple
  - ★ **No Place and Route**

I-Cube's Switch devices use a 100% non-blocking switch matrix. This means any I/O port can be connected to any other I/O port, regardless of other connections in the switch. This is achieved by having a dedicated pass transistor (and associated programming SRAM cell) for all possible pairs of I/O ports.

The switch matrix lines are constructed using a two dimensional wiring structure (using two metal layers) where the horizontal and vertical lines are permanently connected at the "diagonal" points. Thus, for example, horizontal and vertical lines marked "2' are connected together and are in essence a part of the same electrical node. There are pass transistors at the "non-diagonal" intersection points (i.e., at the crossings of horizontal and vertical lines that have different numbers). Notice that there are two possible locations for the pass transistor, either at (i, j) or (j, i) and only one location needs to contain the pass transistor. The locations of the pass transistors are "logically" shown as a triangular array, however the arrangement is quite a bit different on the chip.

This switch matrix structure has several important benefits. The propagation delays are totally predictable. Propagation delays between pin pairs are closely matched and have very little skew. The deterministic architecture also makes the software fast and simple.



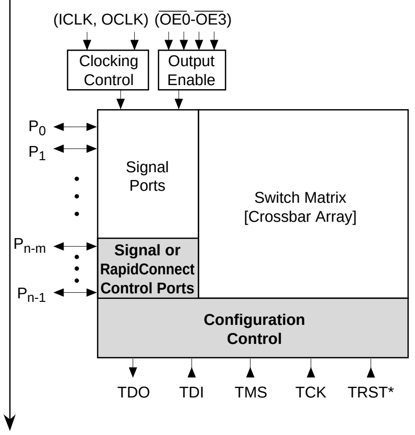
Switch Control / Interface

- ♦ Serial Interface:
  - ★ *(JTAG)*
  - ★ Configure device & make initial connections
  - ★ Ideal for static changes
- ♦ Parallel Interface:
  - ★ *(RapidConfigure)*
  - ★ Gives Users Direct Control of Switch Matrix Resources
  - ★ Ideal For Dynamic and Incremental switching
  - ★ Change (Make or Break) Connections in nanoseconds
  - ★ Can be Driven by CPU or State Machine

I-Cube’s Switch devices use either a serial interface called JTAG or a parallel interface called RapidConfigure. These interfaces are used to program the SRAM cells which control I/O block configuration as well as the switch matrix. The parallel interface is used to directly address the SRAM cells in the switch matrix and makes it possible to change connections very quickly - in a single “write” cycle, which could be as little as 20 ns. On the IQX devices, this interface can also be used for I/O Port configuration. The ability to use this interface for both I/O and switch configuration may eliminate the need for the JTAG interface altogether in certain applications.

The interface uses an address bus, data bus and some control signals and is similar to the CPU to memory interface found in a typical microprocessor based system. Using the address bus you select a switch matrix cell (or multiple cells on some devices) and the data bus contains the data to be written to the cell(s). The control signals are Write Enable (or chip select) and Write Strobe. The width of the address and data bus is different for the different families and devices within a family. RapidConfigure is a write-only interface. It cannot be used to read back the configuration status of the switch or I/O Ports.

IQ Architecture Overview


(ICLK, OCLK) ($\overline{\text{OE0}}$-$\overline{\text{OE3}}$)
Clocking Control
Output Enable
P_0
P_1
P_{n-m}
P_{n-1}
Signal Ports
Switch Matrix [Crossbar Array]
Signal or RapidConnect Control Ports
Configuration Control
TDO
TDI
TMS
TCK
TRST*

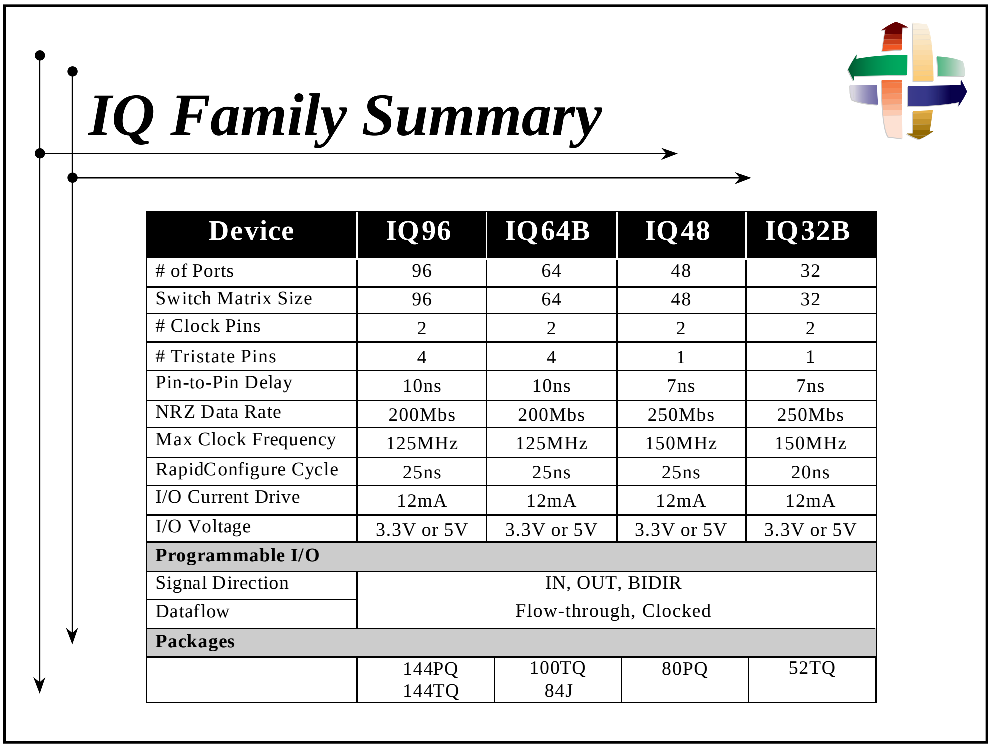
- Switch Matrix
  - Bit-Oriented Switch Matrix
  - 100% Guaranteed Non-Blocking
  - Identical Port-to-Port Delay
- Programmable I/O & Control
  - Input, Output or “Bus Repeater”
  - Flow-through or Clocked Dataflow

The IQ family of bit switching devices was the first family introduced by I-Cube. There are 4 devices based on 0.6µm CMOS process in the family.

The switch matrix in the IQ family is a bit oriented and 100% non-blocking.

The I/O ports are individually programmable for the direction and data flow. There are two global clock lines, one can be used to clock data into the device while the other is to clock data out of the device.

IQ devices **must** use the JTAG interface for initial switch and I/O configuration. For the IQ devices the parallel switch interface is known as “RapidConnect.”



IQ Family Summary

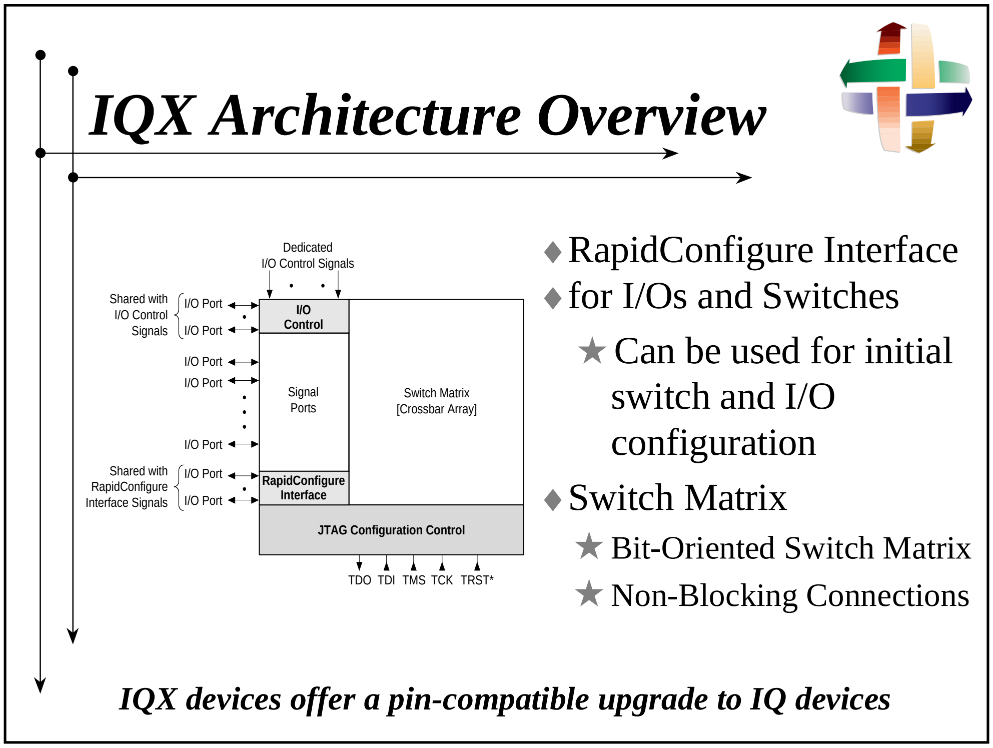
Device	IQ96	IQ64B	IQ48	IQ32B
# of Ports	96	64	48	32
Switch Matrix Size	96	64	48	32
# Clock Pins	2	2	2	2
# Tristate Pins	4	4	1	1
Pin-to-Pin Delay	10ns	10ns	7ns	7ns
NRZ Data Rate	200Mbs	200Mbs	250Mbs	250Mbs
Max Clock Frequency	125MHz	125MHz	150MHz	150MHz
RapidConfigure Cycle	25ns	25ns	25ns	20ns
I/O Current Drive	12mA	12mA	12mA	12mA
I/O Voltage	3.3V or 5V	3.3V or 5V	3.3V or 5V	3.3V or 5V
Programmable I/O				
Signal Direction	IN, OUT, BIDIR			
Dataflow	Flow-through, Clocked			
Packages				
	144PQ 144TQ	100TQ 84J	80PQ	52TQ

The table above summarizes the IQ family devices and features.

Note that the IQ32B-IQ96 devices have separate supply voltages for core and I/O buffers allowing 3.3V I/Os to be supported.

Packages:

J=PLCC, PQ=PQFP, TQ=TQFP



IQX Architecture Overview

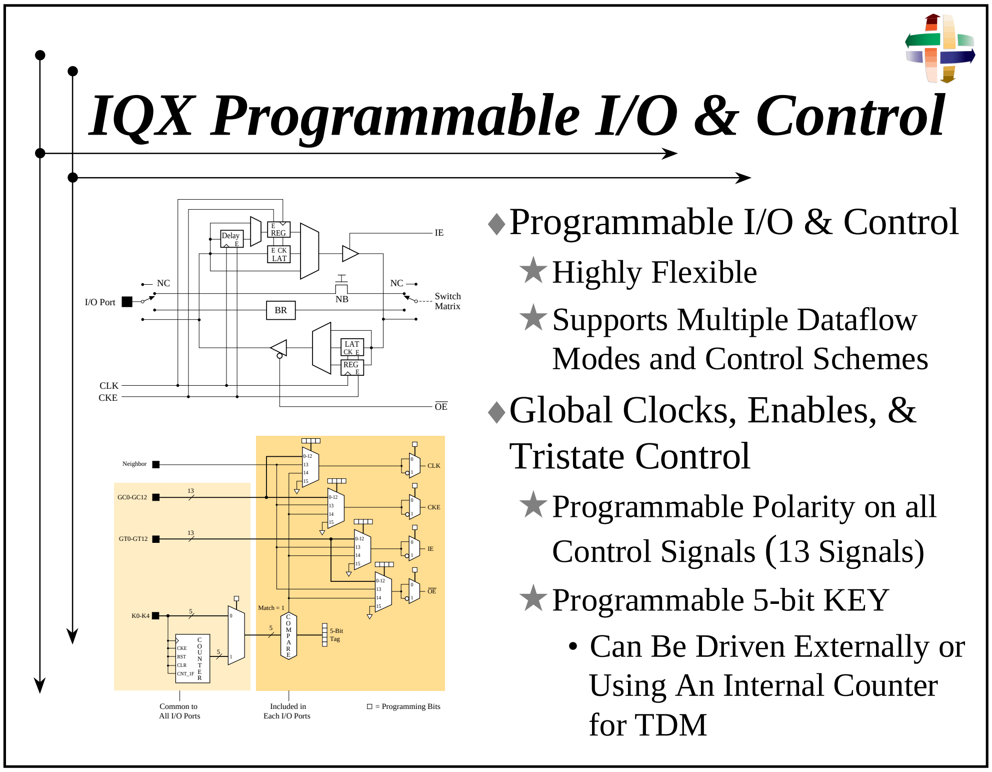
- ♦ RapidConfigure Interface
- ♦ for I/Os and Switches
  - ★ Can be used for initial switch and I/O configuration
- ♦ Switch Matrix
  - ★ Bit-Oriented Switch Matrix
  - ★ Non-Blocking Connections

IQX devices offer a pin-compatible upgrade to IQ devices

IQX is the enhanced bit switching family based on 0.6 µm CMOS process. Four devices are available in the family. The IQX160, IQX128B, IQX320, and IQX240B.

The IQX family uses a bit-oriented switch matrix and programmable I/O ports that are very similar to the I/O ports on the PSX. A small number of I/O ports on the device are dual purpose. They can either be used for signal I/O or for some of the special signals like RapidConfigure signals and I/O control signals.

With the IQX family, the RapidConfigure interface may be used for initial switch and I/O configuration, potentially avoiding the need to use the JTAG interface.



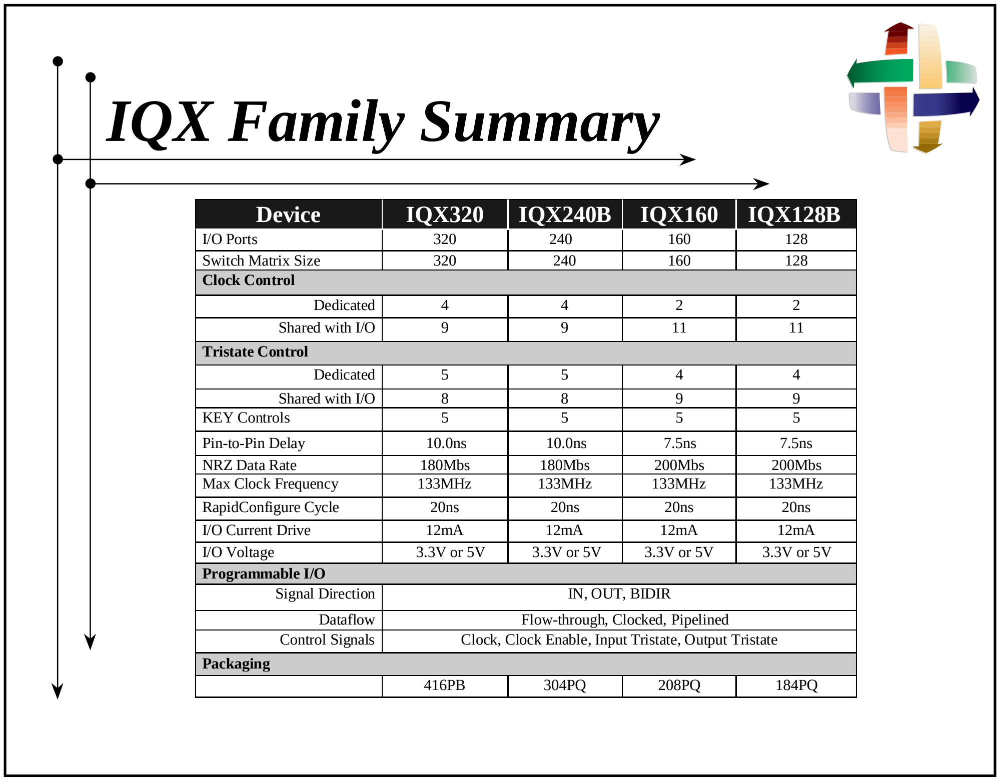
IQX Programmable I/O & Control

- ♦ Programmable I/O & Control
  - ★ Highly Flexible
  - ★ Supports Multiple Dataflow Modes and Control Schemes
- ♦ Global Clocks, Enables, & Tristate Control
  - ★ Programmable Polarity on all Control Signals (13 Signals)
  - ★ Programmable 5-bit KEY
    - Can Be Driven Externally or Using An Internal Counter for TDM

The I/O Ports can be programmed for direction, data flow, and tristatablity. All I/O ports have an identical structure as shown in the bottom figure. There are four control signals, Clock, Clock Enable, Input Enable and Output Enable that control the behavior of each I/O Port.

The source for the control signals is a large pool of pins - some of which are dedicated while some are shared with signal I/Os - that are used to carry the control signals into the device. Each I/O Port can then be individually and independently programmed to use any one or more of these control signals, as shown in the figure on the top. Furthermore, the polarity of these control signals can also be programmed.

In addition to the dedicated control signal sources, a 5-pin “KEY” is used to generate mutually exclusive control signals. Each I/O Port contains a programmable 5-bit tag that is continuously compared against the data on the 5-pin KEY port. A match signal is generated when the key value matches with the programmed tag. This match signal can be used as one additional source for the control signal. The ability to tristate in the input direction allows you to create internal bus structures on the device. These on-chip bus structures can be operated at very high speeds. For example, a combination of this capability and the key feature allows you to build a 32 slot or lower TDM highway that can be operated at 80 MHz clock rates.



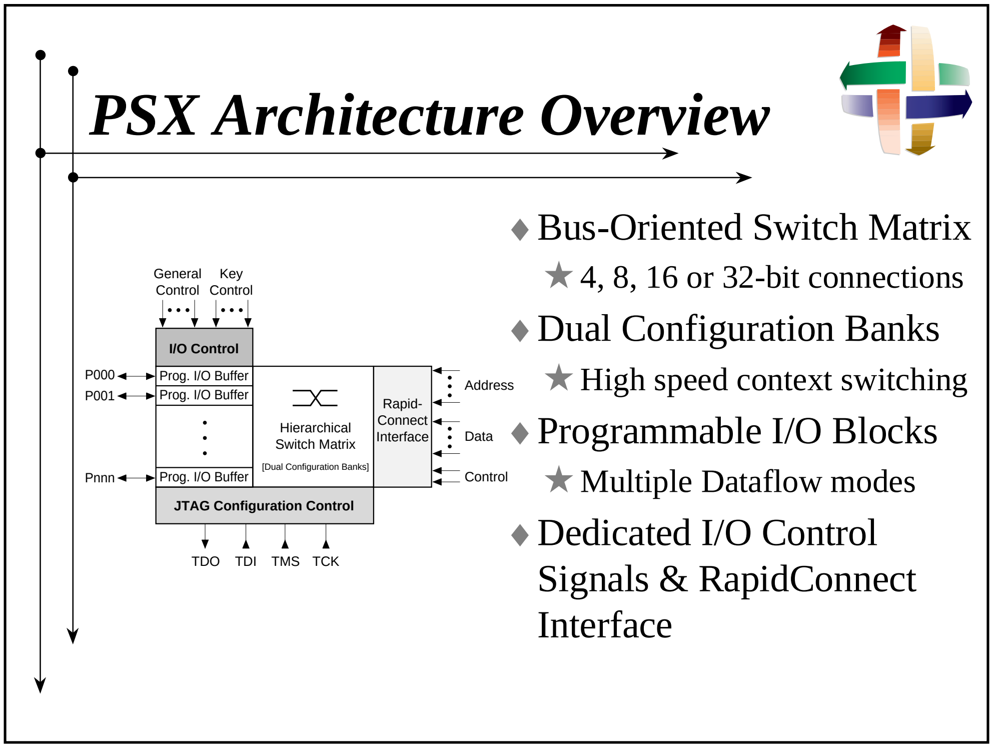
IQX Family Summary

Device	IQX320	IQX240B	IQX160	IQX128B
I/O Ports	320	240	160	128
Switch Matrix Size	320	240	160	128
Clock Control				
Dedicated	4	4	2	2
Shared with I/O	9	9	11	11
Tristate Control				
Dedicated	5	5	4	4
Shared with I/O	8	8	9	9
KEY Controls	5	5	5	5
Pin-to-Pin Delay	10.0ns	10.0ns	7.5ns	7.5ns
NRZ Data Rate	180Mbs	180Mbs	200Mbs	200Mbs
Max Clock Frequency	133MHz	133MHz	133MHz	133MHz
RapidConfigure Cycle	20ns	20ns	20ns	20ns
I/O Current Drive	12mA	12mA	12mA	12mA
I/O Voltage	3.3V or 5V	3.3V or 5V	3.3V or 5V	3.3V or 5V
Programmable I/O				
Signal Direction	IN, OUT, BIDIR			
Dataflow	Flow-through, Clocked, Pipelined			
Control Signals	Clock, Clock Enable, Input Tristate, Output Tristate			
Packaging				
	416PB	304PQ	208PQ	184PQ

The table above summarizes the IQX family devices and features. The IQX320, 240B, 160 and 128B are all in production.

Packages:

PB=PBGA, PQ=PQFP



PSX Architecture Overview

- Bus-Oriented Switch Matrix
  - ★ 4, 8, 16 or 32-bit connections
- Dual Configuration Banks
  - ★ High speed context switching
- Programmable I/O Blocks
  - ★ Multiple Dataflow modes
- Dedicated I/O Control Signals & RapidConnect Interface

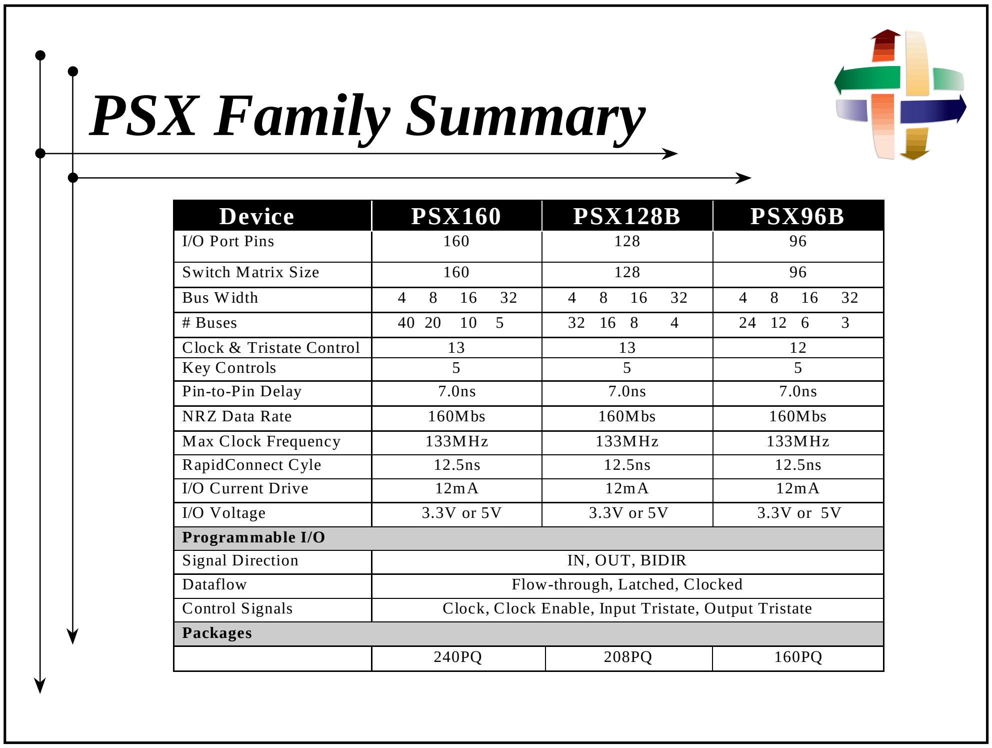
Here is a look at the PSX family. The PSX devices are targeted at dynamic, bus switching applications.

The switch matrix in the PSX is bus-oriented. That means a single SRAM cell in the switch controls the connection status of a group of I/O ports. The minimum granularity or group size is 4 or a nibble.

The I/O ports are very flexible and can be configured in many different ways. The PSX devices have a dedicated pins for RapidConfigure interface and for bringing in a large number of control signals.

Currently, there are three devices in the PSX family. They are all based on 0.6 µm CMOS process.

PSX devices **must** use the JTAG interface for initial switch and I/O configuration. For the PSX devices the parallel switch interface is known as “RapidConnect.”



PSX Family Summary

Device	PSX160	PSX128B	PSX96B
I/O Port Pins	160	128	96
Switch Matrix Size	160	128	96
Bus Width	4 8 16 32	4 8 16 32	4 8 16 32
# Buses	40 20 10 5	32 16 8 4	24 12 6 3
Clock & Tristate Control	13	13	12
Key Controls	5	5	5
Pin-to-Pin Delay	7.0ns	7.0ns	7.0ns
NRZ Data Rate	160Mbs	160Mbs	160Mbs
Max Clock Frequency	133MHz	133MHz	133MHz
RapidConnect Cyle	12.5ns	12.5ns	12.5ns
I/O Current Drive	12mA	12mA	12mA
I/O Voltage	3.3V or 5V	3.3V or 5V	3.3V or 5V
Programmable I/O			
Signal Direction	IN, OUT, BIDIR		
Dataflow	Flow-through, Latched, Clocked		
Control Signals	Clock, Clock Enable, Input Tristate, Output Tristate		
Packages			
	240PQ	208PQ	160PQ

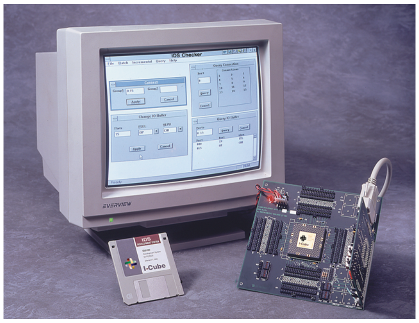
The table above summarizes the PSX family devices and features.

Packages:

PQ=PQFP

Starter (Evaluation) Kits

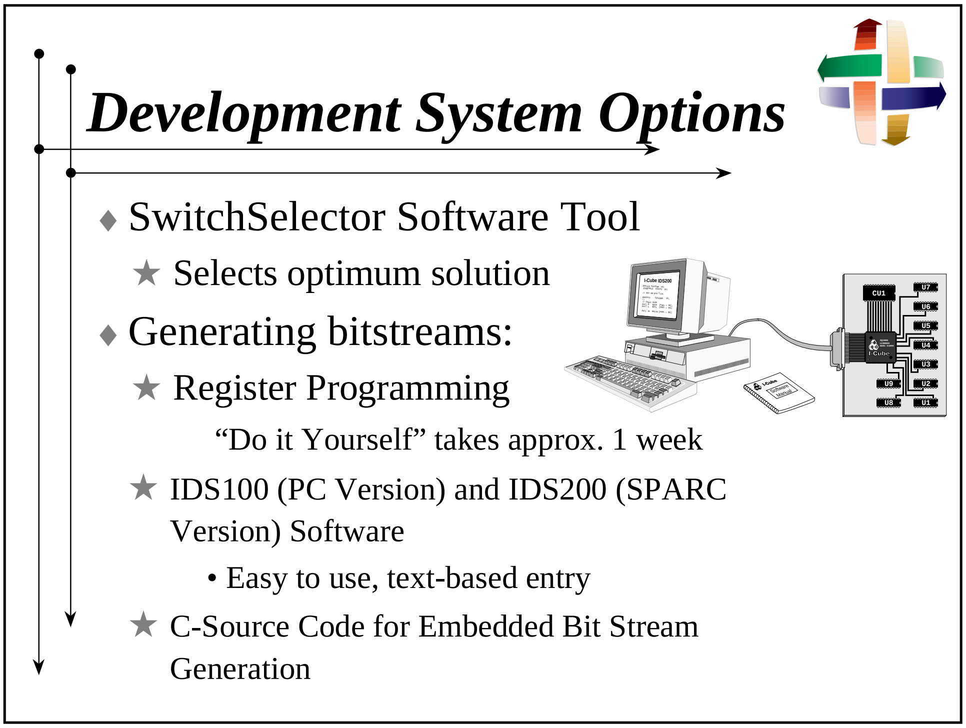
- Complete development tools & applications support to enable easy evaluation & design
- Low Cost Evaluation Kits include :
  - ★ Hardware (PCB, device + Download Cable)
- Interactive Software



I-Cube's crossbar devices and switching technology will likely be 'new' to many engineers. I-Cube has an excellent, low cost starter kit which includes ALL the hardware and software required to start 'playing' with these easy to use devices.

The software runs on a PC (Windows and DOS based) and generates the bitstream required to program or 'configure' the switch. (I-Cube's switches are similar to SRAM FPGAs in needing to be configured at power on).

Starter Kits are available for IQX320, IQX160, IQ96 and PSX160. A cable is supplied to connect the PC to the circuit board via the parallel port to allow the DCS device to be configured. Additionally there is room on the PCB to breadboard a customers own circuit. As the device is SRAM based, it can be reconfigured as many times as required.



Development System Options

- ♦ SwitchSelector Software Tool
  - ★ Selects optimum solution
- ♦ Generating bitstreams:
  - ★ Register Programming

    "Do it Yourself" takes approx. 1 week
  - ★ IDS100 (PC Version) and IDS200 (SPARC Version) Software
    - • Easy to use, text-based entry
  - ★ C-Source Code for Embedded Bit Stream Generation

CU1
U7
U6
U5
U4
U3
U2
U1
U9
U8

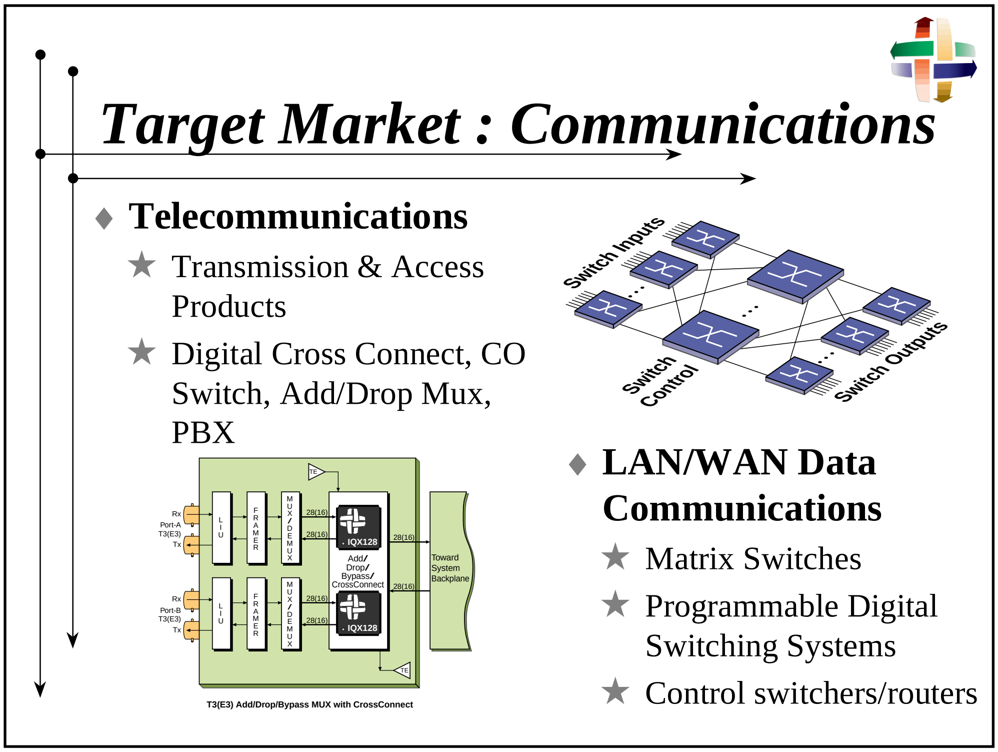
I-Cube offers a range of low cost tools to allow engineers to create designs with switch-based architectures. The SwitchSelector software tool accepts inputs such as # inputs, # outputs, bus width, and then provide guidance on how to implement the switch using one or multiple of I-Cube's switch devices. This allows the engineer to optimize his design for minimum cost, lowest device count, smallest PCB area, etc.

As a next step, for easy, low cost and risk evaluation I-Cube offers Starter Kits which include all the software and hardware needed to build a switch.

There are several different ways to generate the bit stream for configuring I-Cube's switch devices. The optimal choice will be different for different applications.

The first option is called off-line bit stream generation. This method is suited for applications requiring static or pseudo-static interconnect or routing. Under this option, the necessary bit stream(s) are generated ahead of time. They can then be stored in a non-volatile memory like flash or EPROM or they could be downloaded from a host such as a PC.

The **I**-Cube **D**evelopment **S**ystem (IDS) software is an easy to use tool which supports ALL I-Cube devices. We offer two products that allow the user to generate the bit stream - PC/Windows product called the **IDS100** and SparcStation version called the **IDS200**.



Target Market : Communications

- **Telecommunications**
  - ★ Transmission & Access Products
  - ★ Digital Cross Connect, CO Switch, Add/Drop Mux, PBX

Switch Inputs
Switch Control
Switch Outputs
Rx
Port-A
T3(E3)
Tx
Rx
Port-B
T3(E3)
Tx
LIU
FRAMER
MUX/DEMUX
TE
28(16)
IQX128
Add/
Drop/
Bypass/
CrossConnect
Toward
System
Backplane

T3(E3) Add/Drop/Bypass MUX with CrossConnect

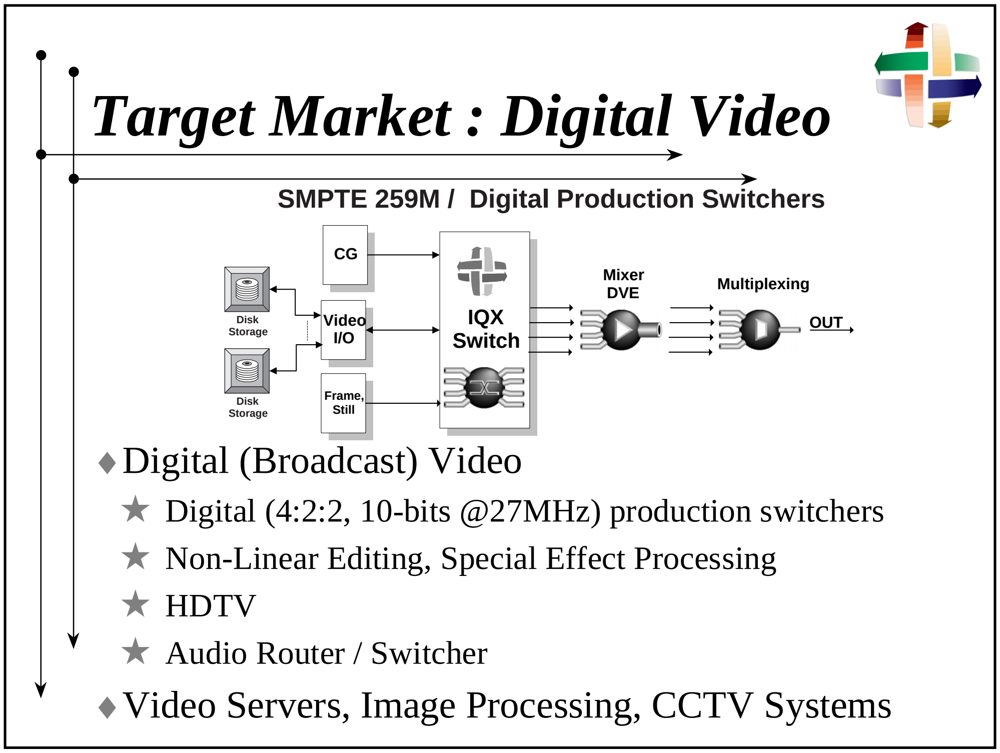
- **LAN/WAN Data Communications**
  - ★ Matrix Switches
  - ★ Programmable Digital Switching Systems
  - ★ Control switchers/routers

I-Cube's switching technology is a great fit in WAN, Digital Cross Connect, and Digital Access Products, performing functions such as crosspoint switching, signal routing, data aggregation and switching. Additionally PSIDs may be used to design programmable backplanes or midplanes.

I-Cube's IQX family are a great solution for all matrix switches, both single stage and multiple stage (e.g. CLOS networks) as found in applications such as Digital Cross Connects (DCC or DCS), LAN/WAN switches, RS232/422 data switches, etc.

The IQX products provide the ability to build a fully non-blocking solution supporting multicast or broadcast, with the benefits of higher performance, tighter skew, low pulse width distortion and lower device count.

I-Cube's SwitchSelector gives the user options on how to implement a given size of switch matrix with various IQX or IQ products, optimizing for minimum cost or minimum PCB area.



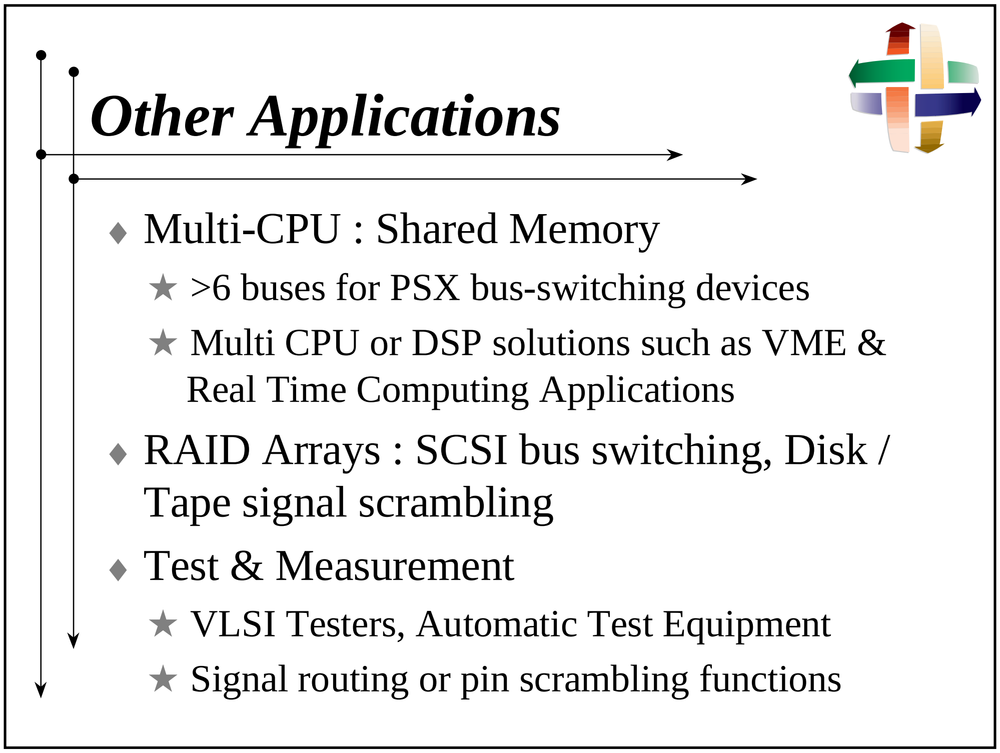
Target Market : Digital Video

SMPTE 259M / Digital Production Switchers

- Digital (Broadcast) Video
  - Digital (4:2:2, 10-bits @27MHz) production switchers
  - Non-Linear Editing, Special Effect Processing
  - HDTV
  - Audio Router / Switcher
- Video Servers, Image Processing, CCTV Systems

I-Cube switching products are a great fit for digital video switching applications such as video servers, video editing, broadcast video switching, audio routing and image processing.

DCS products provide the ideal solution for multiplexing and routing high bandwidth video buses (4:2:2), as well as digital audio switching. With a range of devices offering from 128 to 320 I/O’s, IQX devices provide the designer the flexibility of using a single chip or multiple chip solution. For a matrix with a large number of 10-bit buses to switch, a cost effective solution can be implemented by “bit slicing” the bus across multiple IQX devices which simplifies the switching control and reduces system cost.

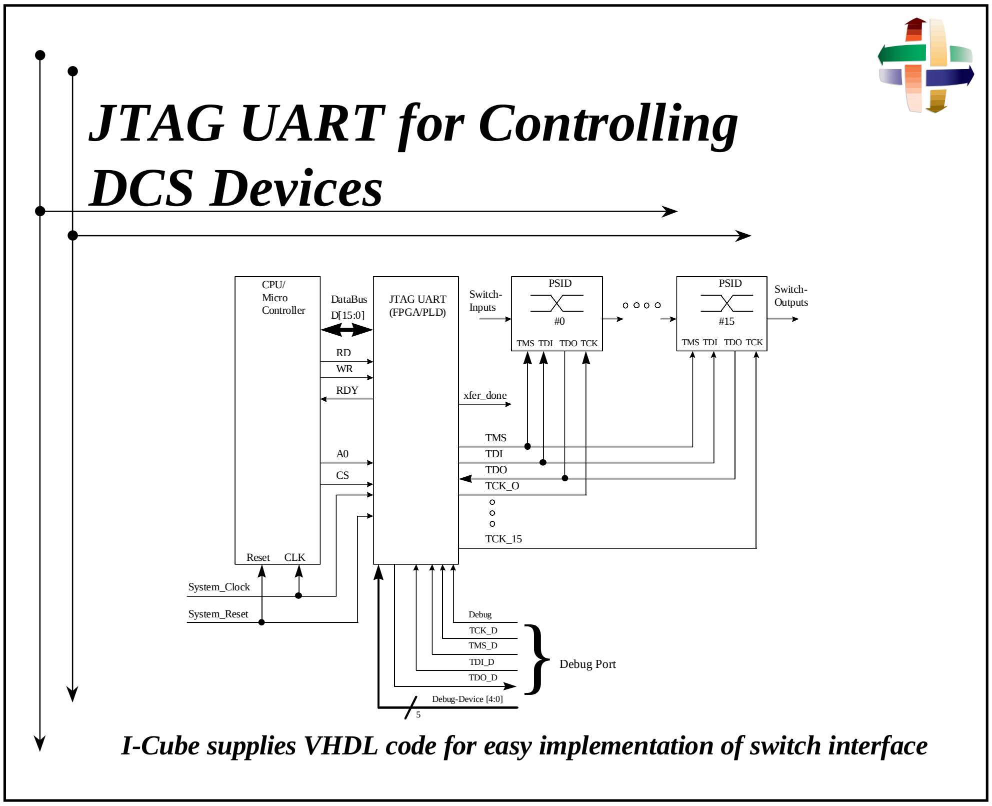


Other Applications

- Multi-CPU : Shared Memory
  - ★ >6 buses for PSX bus-switching devices
  - ★ Multi CPU or DSP solutions such as VME & Real Time Computing Applications
- RAID Arrays : SCSI bus switching, Disk / Tape signal scrambling
- Test & Measurement
  - ★ VLSI Testers, Automatic Test Equipment
  - ★ Signal routing or pin scrambling functions

For high performance computing applications I-Cube’s switch products offer an ideal solution for high performance bus switching and flexible backplanes.

For test and measurement applications, DCSs are an excellent solution for signal routing, pin scrambling, pattern generation, and other “programmable” functions.



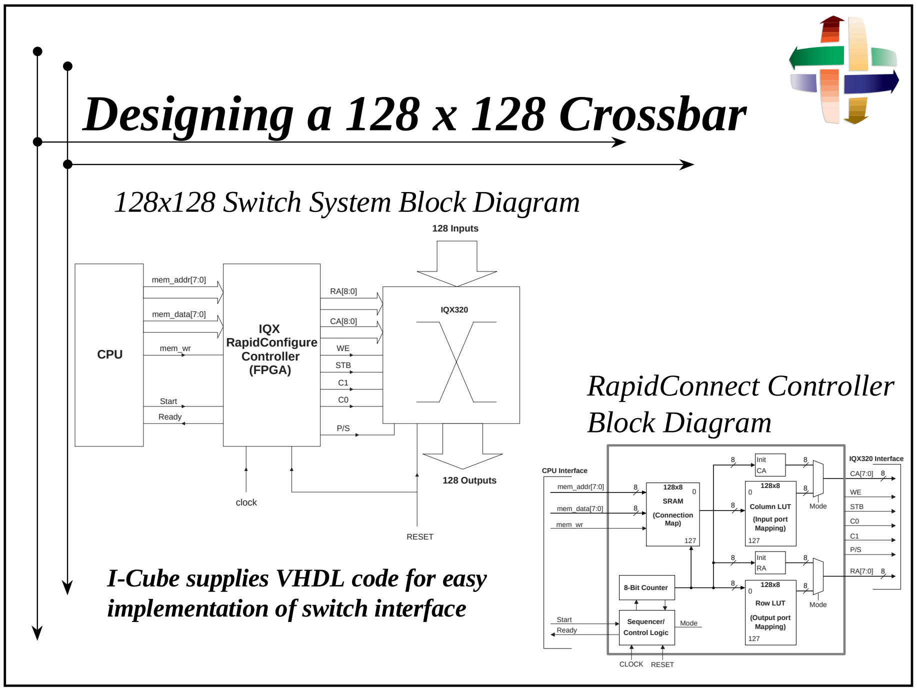
JTAG UART for Controlling DCS Devices

I-Cube supplies VHDL code for easy implementation of switch interface

I-Cube’s Digital Crosspoint Switching (DCS) Devices have a serial interface to allow configuration of the I/O ports and switch matrix. This serial interface is compliant to the IEEE1149.1 standard, commonly referred to as JTAG. (The advantage in using the JTAG interface for controlling a DCS device, as opposed to using the parallel or RapidConfigure interface, is that it allows all the I/O Ports on the IQ and IQX devices to be used for switch inputs or outputs.)

This application note presents a design example showing how to build a “JTAG UART” using an FPGA/CPLD. This allows an easy interface between a microcontroller with an 8 or 16-bit data bus and the JTAG interface of a DCS device. This example allows up to 16 separate JTAG interfaces to be supported. Additionally, the JTAG UART includes support for a debug mode, which allows the user to connect the target DCS device to the parallel port of a PC, and to control it using I-Cube’s IDS100 development software running on the PC.

Note : the use of the JTAG port for power on configuration is mandatory with the IQ and PSX families, but is optional with the IQX family.



Designing a 128 x 128 Crossbar

128x128 Switch System Block Diagram

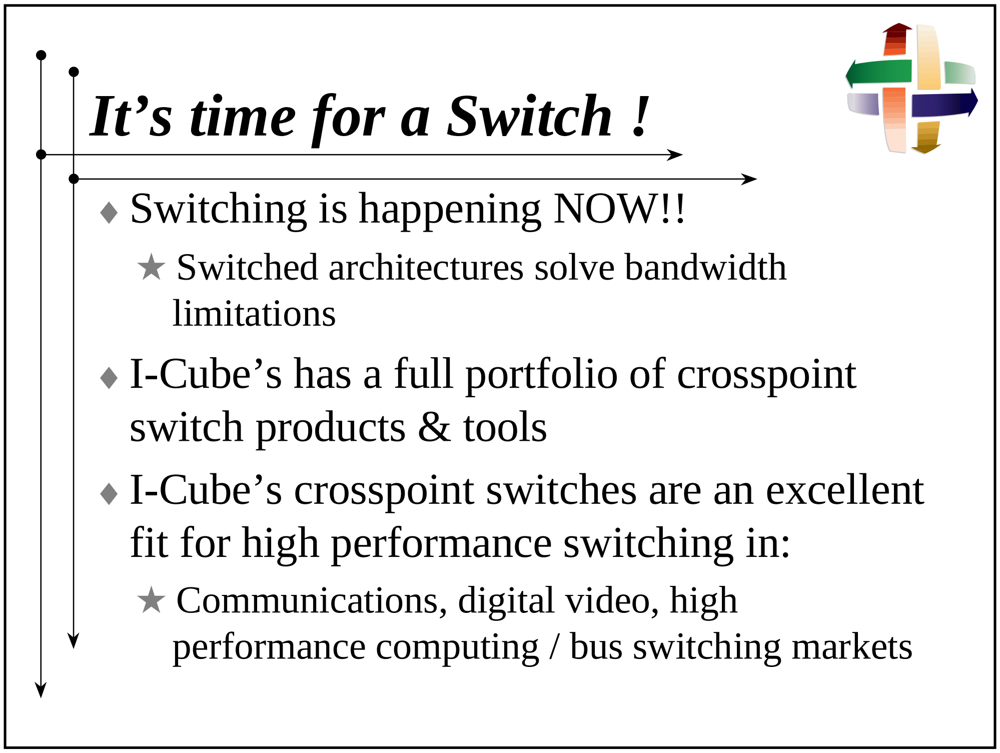
RapidConnect Controller Block Diagram

I-Cube supplies VHDL code for easy implementation of switch interface

This application note describes how to implement a 128x128 crossbar using an IQX320 device. A complete design is detailed including how to implement the RapidConfigure interface which is used for fast reconfiguration. A FPGA/CPLD based controller can be used to implement the RapidConfigure (RC) interface on the IQX320 device, and the VHDL code to implement such a controller is provided.

The block diagram for the switch controller is shown above. The switch controller consists of a single or dual port SRAM memory, 128 x 8 that stores the connection information, organized by the outputs in the switch. In addition to the SRAM, the switch controller also has two 128 x 8 ROM LUTs (Look Up Tables) which gets initialized at power-on. These ROM tables contain port mapping information to map the user I/O ports to the actual IQX320 ports. ROM_RA maps the switch output ports (i.e. OUT_0 to OUT_127) to the corresponding I/O ports used as outputs for the IQX320. The ROM_RA data eventually becomes the Row Addresses RA[7:0] for the RapidConfigure interface. ROM_CA maps the switch input ports (IN_0 to IN_127) to the corresponding I/O ports used as inputs for the IQX320. The ROM_CA data eventually becomes the Column Addresses CA[7:0] for the RapidConfigure interface. User port is used as ROM address and the data corresponds to the actual IQX320 port.

The reconfiguration for the complete crossbar switch is to be accomplished in under 10 µs. This is a synchronous design allowing easy implementation in FPGA/CPLD.



It’s time for a Switch !

- Switching is happening NOW!!
  - ★ Switched architectures solve bandwidth limitations
- I-Cube’s has a full portfolio of crosspoint switch products & tools
- I-Cube’s crosspoint switches are an excellent fit for high performance switching in:
  - ★ Communications, digital video, high performance computing / bus switching markets

Switching-technology is set to follow a rapid adoption curve; already innovators, especially in telecommunication and networking applications, have been using switching technology for many years.

Now with the availability of high performance standard products from I-Cube the ***mainstream*** users will be able to take advantage of the many benefits offered by switch-based architectures.

DCS devices are ideally suited to perform the following functions and offer a great deal of value to the system designer for the following applications.

Crosspoint switching applications, particularly those requiring a large switching element.

DCS devices (IQX and PSX) can be used for constructing **multiplexers and demultiplexers.**

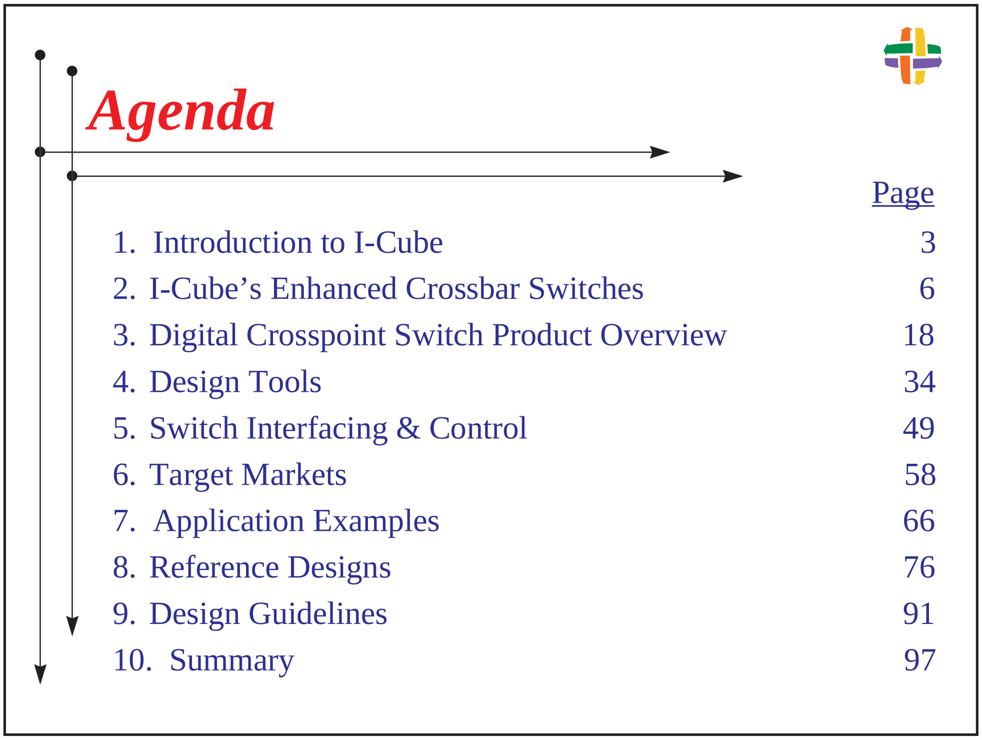
Data aggregation / concentration applications requiring a large number of incoming channels (16 or more) or wide channels.

Static routing applications requiring routing among a large number of signals.

Digital Crosspoint Switching Solutions

I-Cube, Inc.

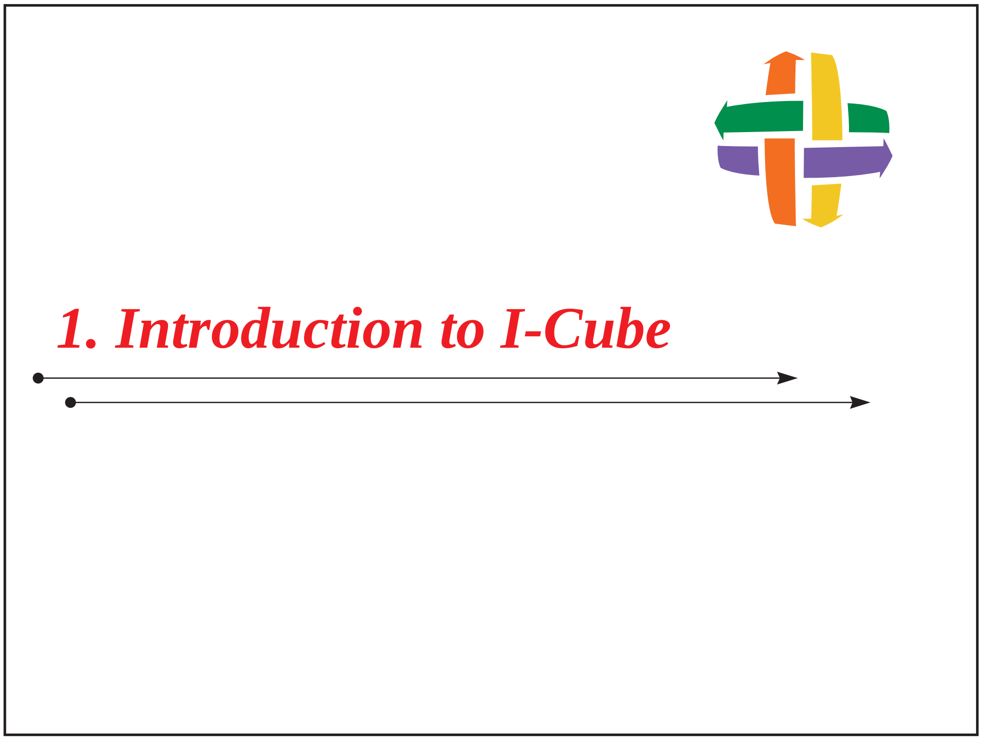
I-Cube, Inc.
2605 S. Winchester Blvd.
Campbell, CA 95008
Tel: (408) 341 1888
Fax: (408) 341 1899
Web site: www.icube.com



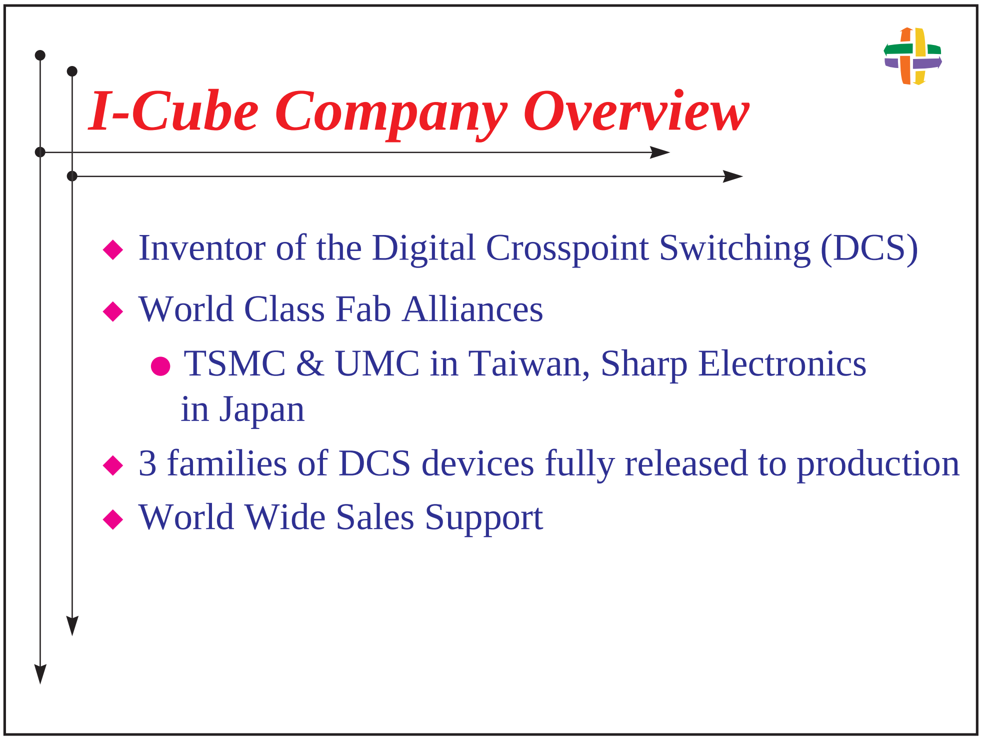
Agenda

	Page
1. Introduction to I-Cube	3
2. I-Cube’s Enhanced Crossbar Switches	6
3. Digital Crosspoint Switch Product Overview	18
4. Design Tools	34
5. Switch Interfacing & Control	49
6. Target Markets	58
7. Application Examples	66
8. Reference Designs	76
9. Design Guidelines	91
10. Summary	97

This seminar will cover an introduction to switching technology, I-Cube’s products and design tools, and a range of application examples showing how to design and build high performance switch-based systems.



1. Introduction to I-Cube



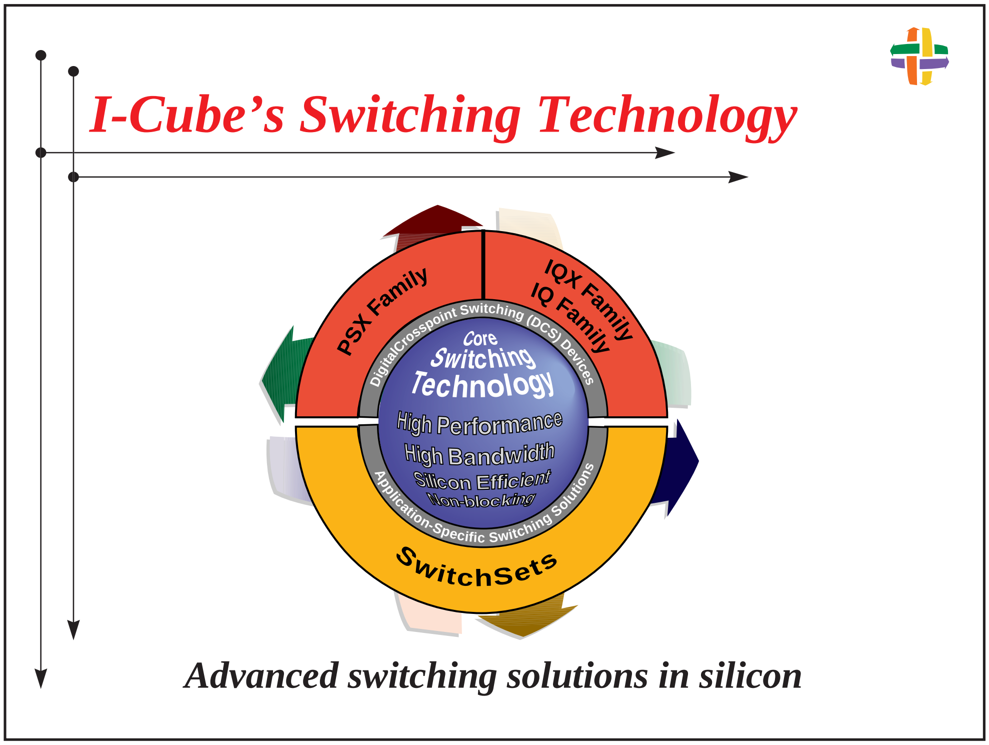
I-Cube Company Overview

- Inventor of the Digital Crosspoint Switching (DCS)
- World Class Fab Alliances
  - TSMC & UMC in Taiwan, Sharp Electronics in Japan
- 3 families of DCS devices fully released to production
- World Wide Sales Support

I-Cube's high performance, non-blocking switching technology provides cost effective bandwidth management solutions for high performance systems.

I-Cube is a privately held company headquartered in Campbell, CA.

I-Cube's products are designed in-house and manufactured using the leading edge mainstream process technology offered by selected foundry partners. Devices are assembled and tested using multiple domestic and off-shore sources. I-Cube uses a network of domestic distributor and sales reps, and international stocking reps.

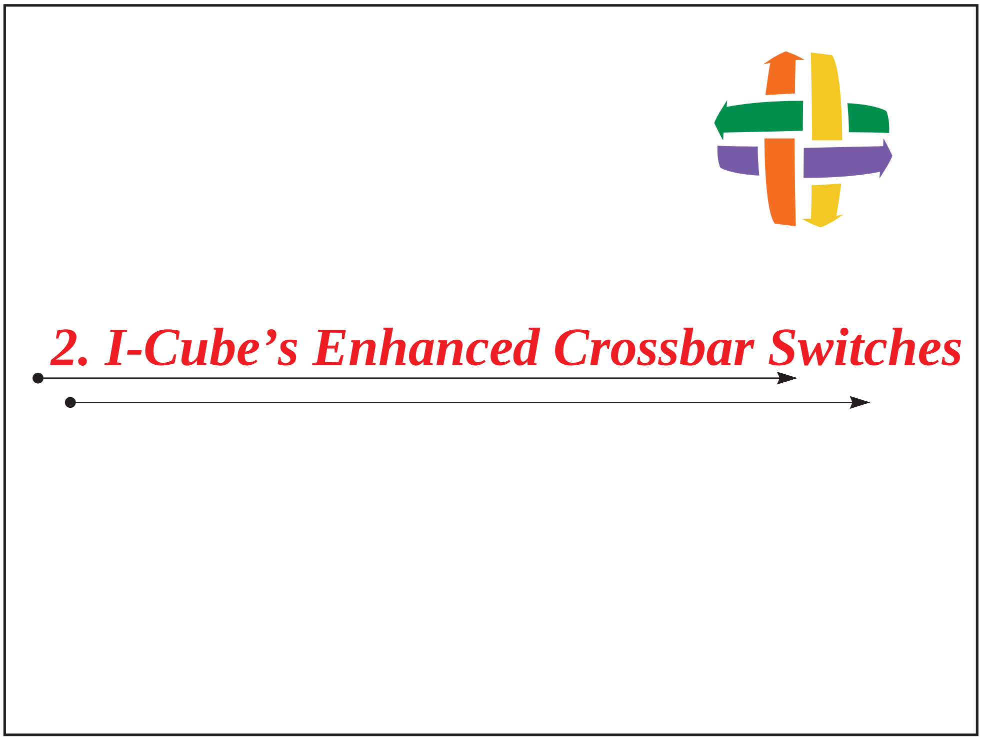


I-Cube's Switching Technology

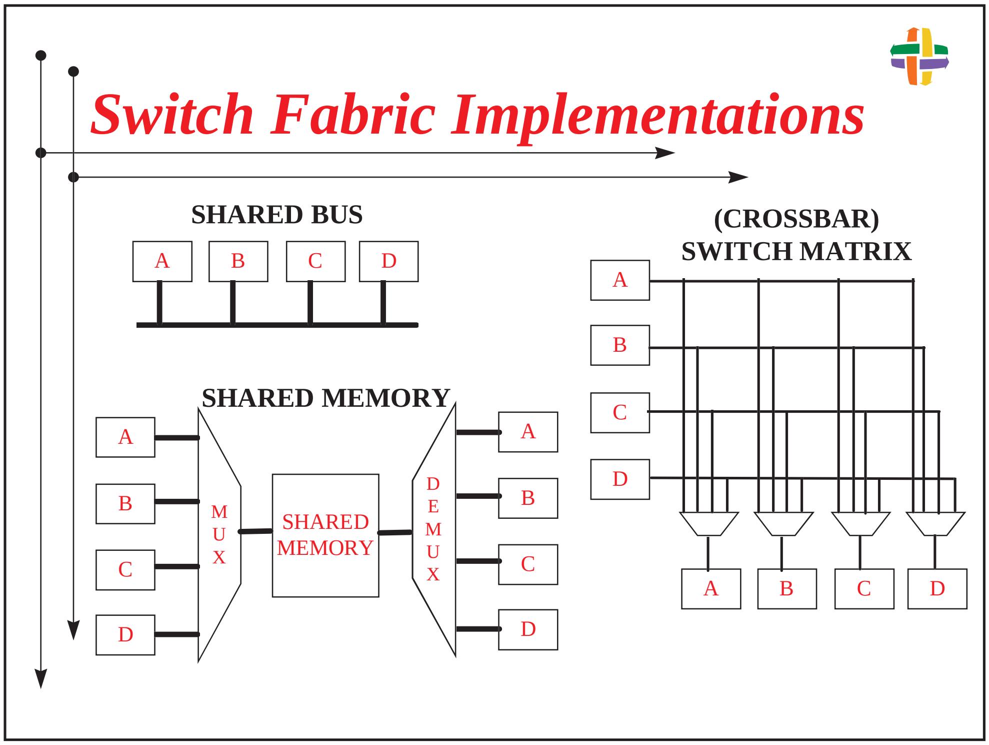
Advanced switching solutions in silicon

I-Cube's mission is to develop semiconductor solutions for digital switching applications based on the company's proprietary, high performance switching technology. The heart of I-Cube's products is the non-blocking, high performance silicon switching technology, which is heavily patented. The company has leveraged its core switching technology with both it's generic Digital Crosspoint Switching (DCS) devices and application specific chip sets (SwitchSets™).

DCS devices are used in a wide range of applications including: telecommunications, networking, and digital video/imaging applications. SwitchSets are targeted at LAN/WAN and Gigabit switching.



2. I-Cube's Enhanced Crossbar Switches



Switch Fabric Implementations

SHARED BUS

A B C D

SHARED MEMORY

A B C D

MUX

SHARED MEMORY

DEMUX

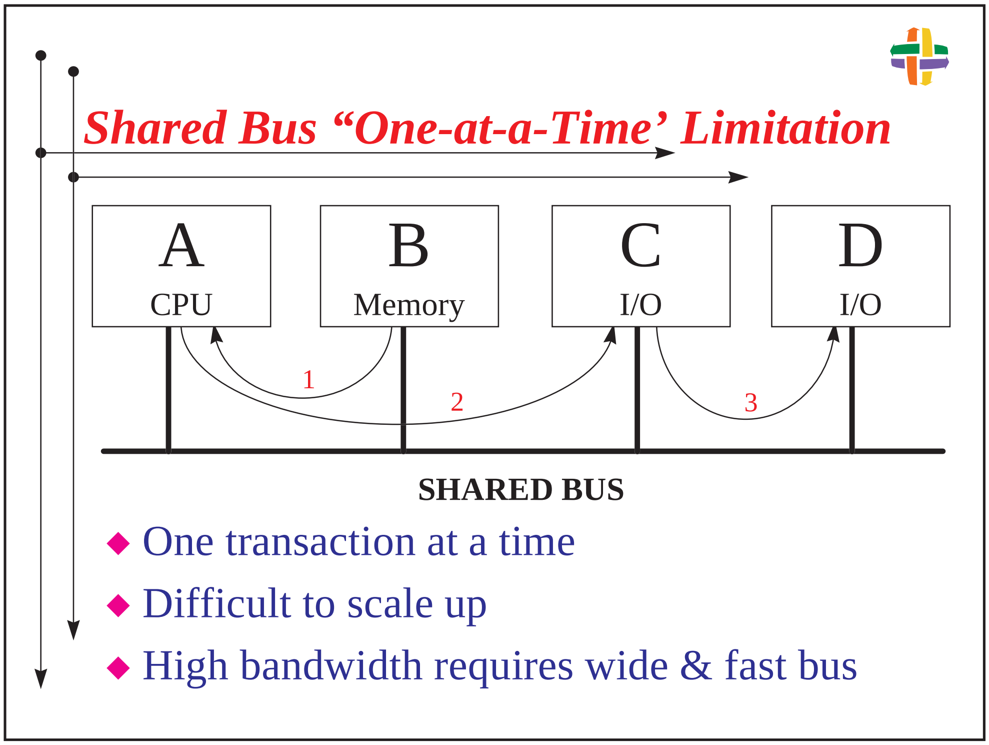
A B C D

(CROSSBAR) SWITCH MATRIX

A B C D

A B C D

The three most common ‘switch fabrics’ are shown above. The next three pages look at the relative advantages and disadvantages of each solution.



Shared Bus "One-at-a-Time' Limitation

- One transaction at a time
- Difficult to scale up
- High bandwidth requires wide & fast bus

BUS

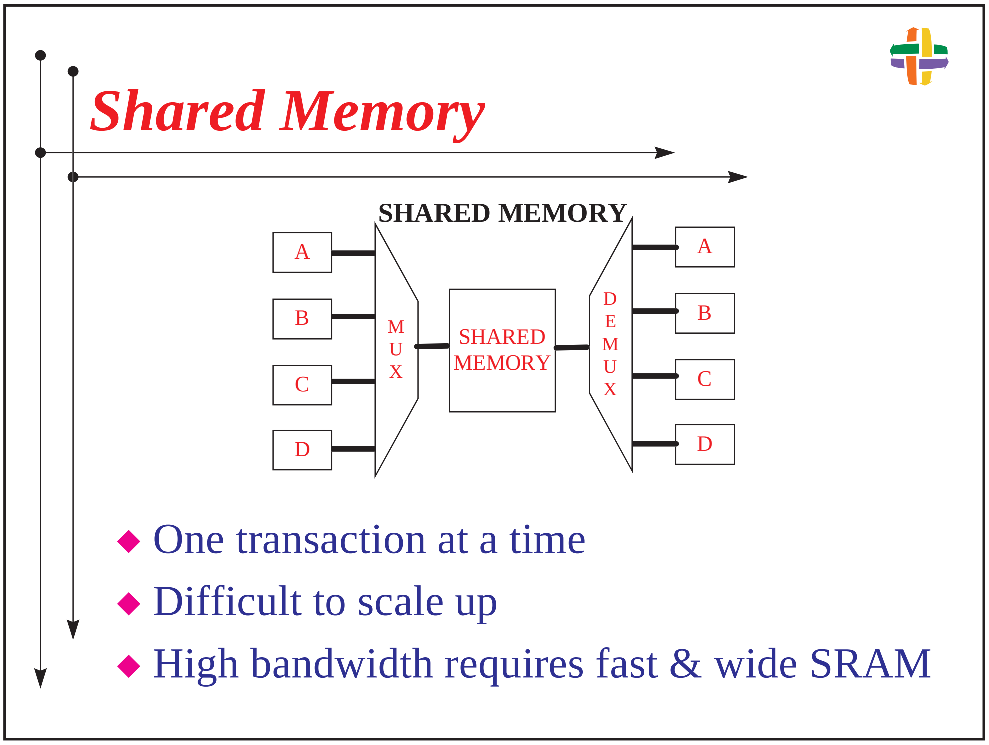
•Easy to understand, components readily available

•For higher bandwidth you:

(a) clock the bus faster, or (b) have a wider datapath

•Running into the limits of practical performance (physics and cost)

A shared bus has the limitation that only one 'master' can drive data onto the bus at a time (although multiple 'listeners' can take data from the bus) is the basic limitation of 'one-at-a-time' data transfer.



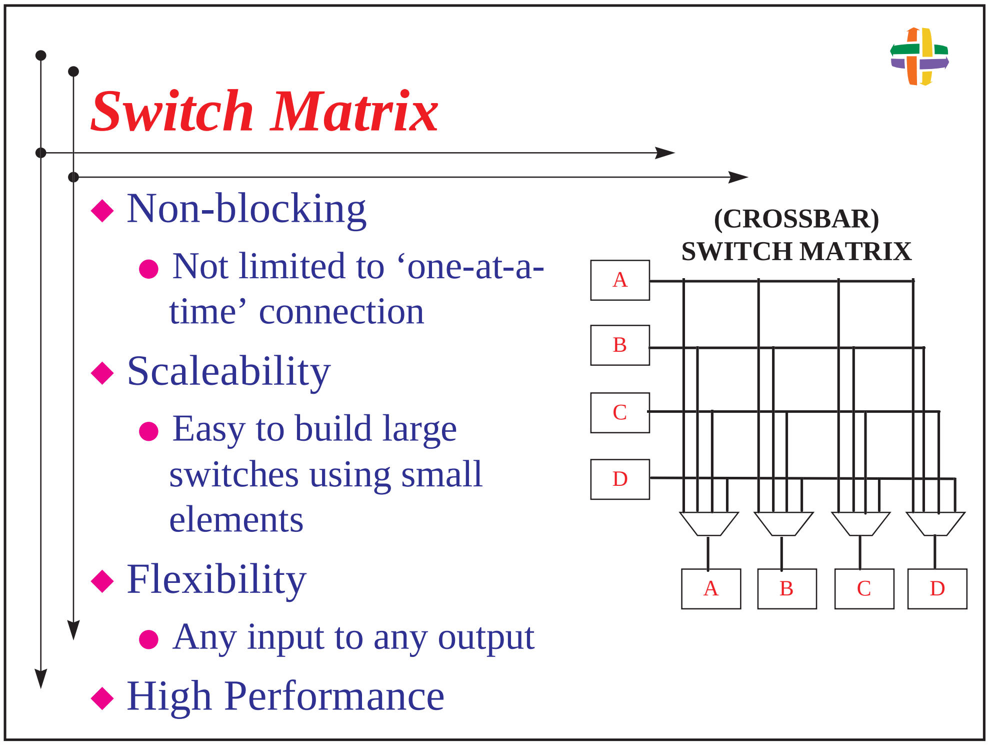
Shared Memory

- One transaction at a time
- Difficult to scale up
- High bandwidth requires fast & wide SRAM

<u>SHARED MEMORY</u>

•Multiple processor single memory or multiple processor multiple memory topology

•High performance requires fast memory (expensive) and high I/O mux/demux logic (typically requires custom ASIC)



Switch Matrix

- ◆ Non-blocking
  - ● Not limited to ‘one-at-a-time’ connection
- ◆ Scaleability
  - ● Easy to build large switches using small elements
- ◆ Flexibility
  - ● Any input to any output
- ◆ High Performance

(CROSSBAR)
SWITCH MATRIX
A
B
C
D
A
B
C
D

SWITCH MATRIX

•Guaranteed High Performance …

* Non-blocking

•Not limited to ‘one-at-a-time’ connection

* Flexibility

•Any input to any output connections possible

•Single input can be connected to multiple outputs (known as multicast or broadcast

•Rearrangeable

* Scaleability

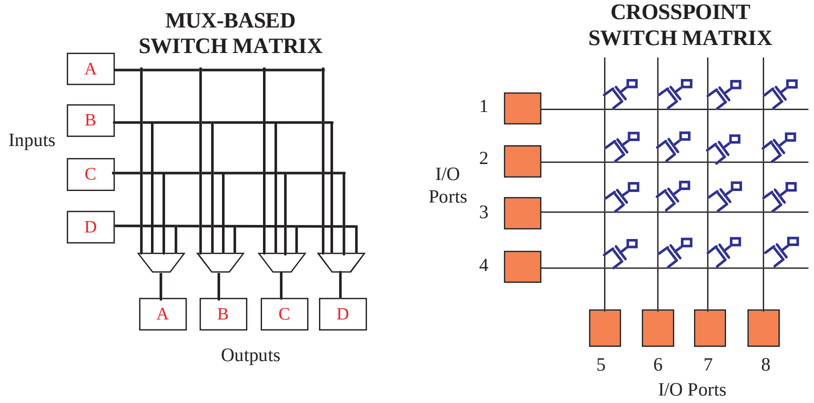
•Easy to build large switches using small elements

•Crossbars USED to be regarded as expensive, required high I/O count for large networks, and difficult to design to achieve high performance

•I-Cube’s switches solve previous limitations with flexible I/O’s supporting bi-directional interconnect in a silicon efficient, cost effective CMOS process

Crossbar Matrices

- There are 2 ways to build a crossbar matrix :



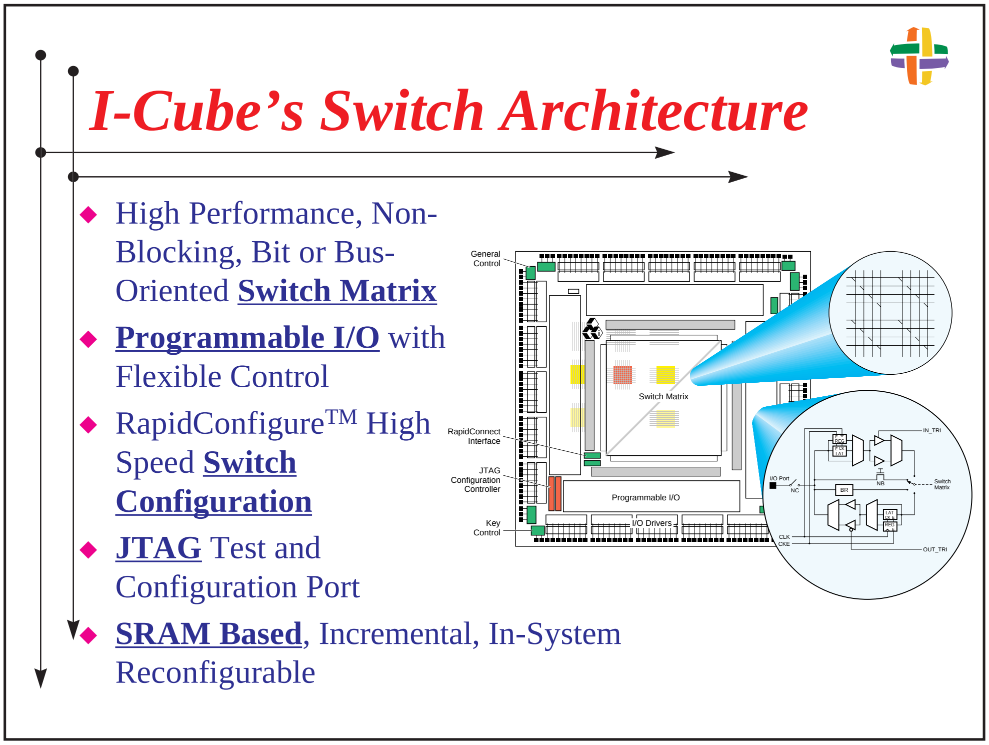
Crossbar switching arrays can be implemented in one of two ways:

1. Using a **N-way multiplexer** (see left hand diagram above) at each output port to select data from the input ports

•This is the traditional way semiconductor vendors have built crossbar switch products, and while it is easy to understand it has major limitations for architectural flexibility/ performance, and efficient implementation in silicon.

2. A **crosspoint array** (see right hand diagram above) with a switching element at each input/output intersection.

•This is the method I-Cube’s crosspoint switches use to offer a high density crosspoint array which gives greater flexibility than traditional crossbar switches making it feasible to build large switches in a cost-effective way using modern CMOS technology.



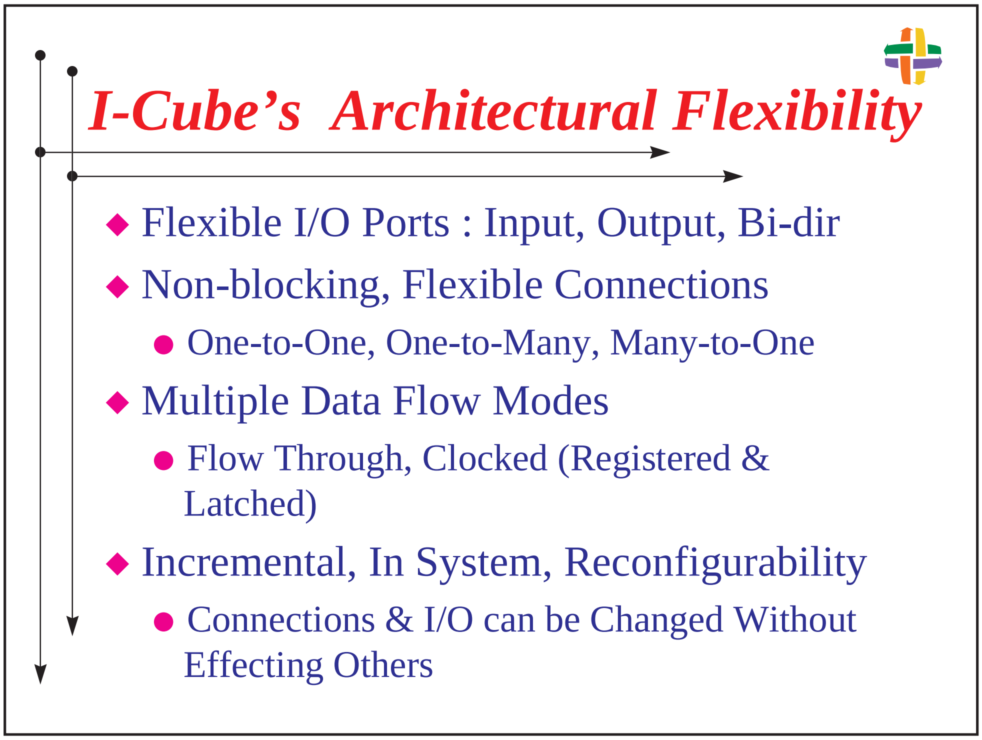
I-Cube's Switch Architecture

- High Performance, Non-Blocking, Bit or Bus-Oriented **Switch Matrix**
- **Programmable I/O** with Flexible Control
- RapidConfigure^{TM} High Speed **Switch Configuration**
- **JTAG** Test and Configuration Port
- **SRAM Based**, Incremental, In-System Reconfigurable

The architecture of the crosspoint switching devices consists of four basic blocks;

1. The switch matrix
2. The programmable I/O blocks
3. The configuration and switch control
4. The clocking and tri-state controls (port control signals)

Each programmable I/O port on the switching device is connected to a unique line in the switch matrix. Digital signals are brought into the device through I/O ports that are configured as inputs, the signals are then switched using the switch matrix, and they go out over the I/O ports that are configured as outputs. These devices use SRAM based configuration and therefore must be configured in-system every time the power is turned on. The primary mode for configuration is the JTAG serial interface. A parallel interface called the RapidConfigure interface allows quick reconfiguration of the switch matrix and, in the case of the new IQX family, programmable I/O ports as well.



I-Cube's Architectural Flexibility

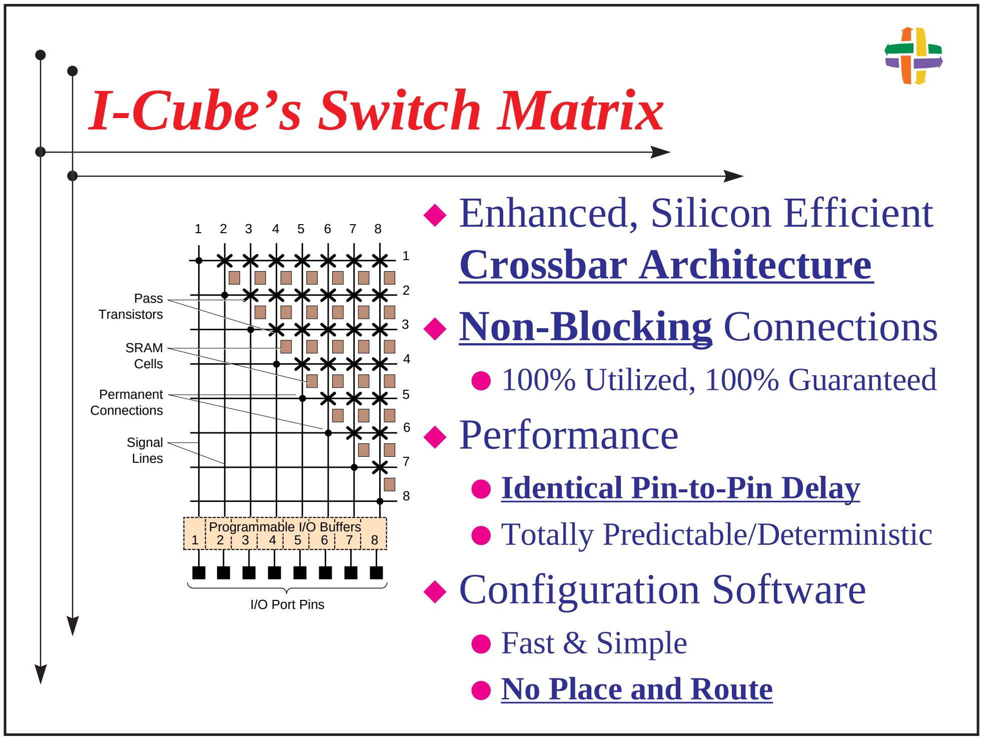
- ◆ Flexible I/O Ports : Input, Output, Bi-dir
- ◆ Non-blocking, Flexible Connections
  - ● One-to-One, One-to-Many, Many-to-One
- ◆ Multiple Data Flow Modes
  - ● Flow Through, Clocked (Registered & Latched)
- ◆ Incremental, In System, Reconfigurability
  - ● Connections & I/O can be Changed Without Effecting Others

One of the key advantages offered by I-Cube's switching devices is the flexibility they offer the system designer. The flexibility manifests itself in multiple ways; the I/Os on the device are identical and user programmable. You can make them into input, output or bidirectional. You can make 1-to-1 (point-to-point), 1-to-many (point-to-multi point) or many-to-one connections through the switching core. The data flowing through the device can be flow through, registered or latched - this in combination with programmable clocks and clock polarity can be used very effectively in high performance designs where the margins on set up and hold times are very tight.

•The device can be configured incrementally so you can alter I/O port & connection configurations without affecting data integrity on any "live" paths in the device.

•Depending on the application, there are multiple configuration options to choose from. Off-line or in-system bit stream generation, using I-Cube supplied or customer developed software.

•The JTAG interface may be used for board-level testing as well as device configuration. It can also be used to readback the configuration status of switch matrix and I/O ports.



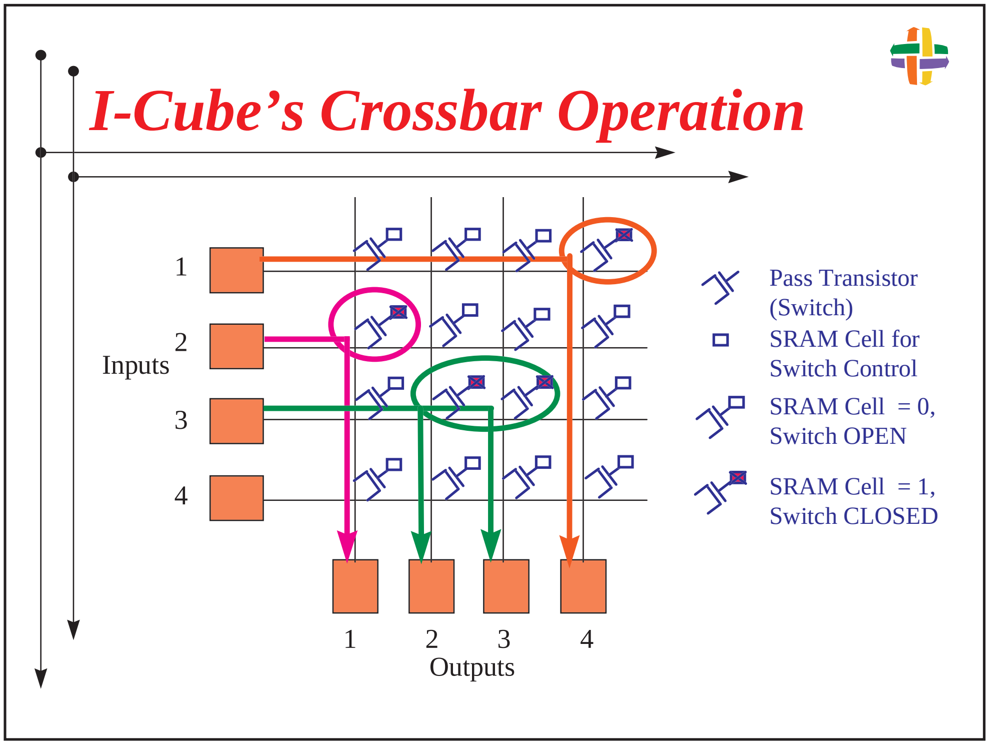
I-Cube’s Switch Matrix

- Enhanced, Silicon Efficient **Crossbar Architecture**
- **Non-Blocking** Connections
  - 100% Utilized, 100% Guaranteed
- Performance
  - **Identical Pin-to-Pin Delay**
  - Totally Predictable/Deterministic
- Configuration Software
  - Fast & Simple
  - **No Place and Route**

I-Cube’s Switch devices use a 100% non-blocking switch matrix. This means any I/O port can be connected to any other I/O port, regardless of other connections in the switch. This is achieved by having a dedicated pass transistor (and associated programming SRAM cell) for all possible pairs of I/O ports.

The switch matrix lines are constructed using a two dimensional wiring structure (using two metal layers) where the horizontal and vertical lines are permanently connected at the “diagonal” points. Thus, for example, horizontal and vertical lines marked “2’ are connected together and are in essence a part of the same electrical node. There are pass transistors at the “non-diagonal” intersection points (i.e., at the crossings of horizontal and vertical lines that have different numbers). Notice that there are two possible locations for the pass transistor, either at (i, j) or (j, i) and only one location needs to contain the pass transistor. The locations of the pass transistors are “logically” shown as a triangular array, however the arrangement is quite a bit different on the chip.

This switch matrix structure has several important benefits. The propagation delays are totally predictable. Propagation delays between pin pairs are closely matched and have very little skew. The deterministic architecture also makes the software fast and simple.



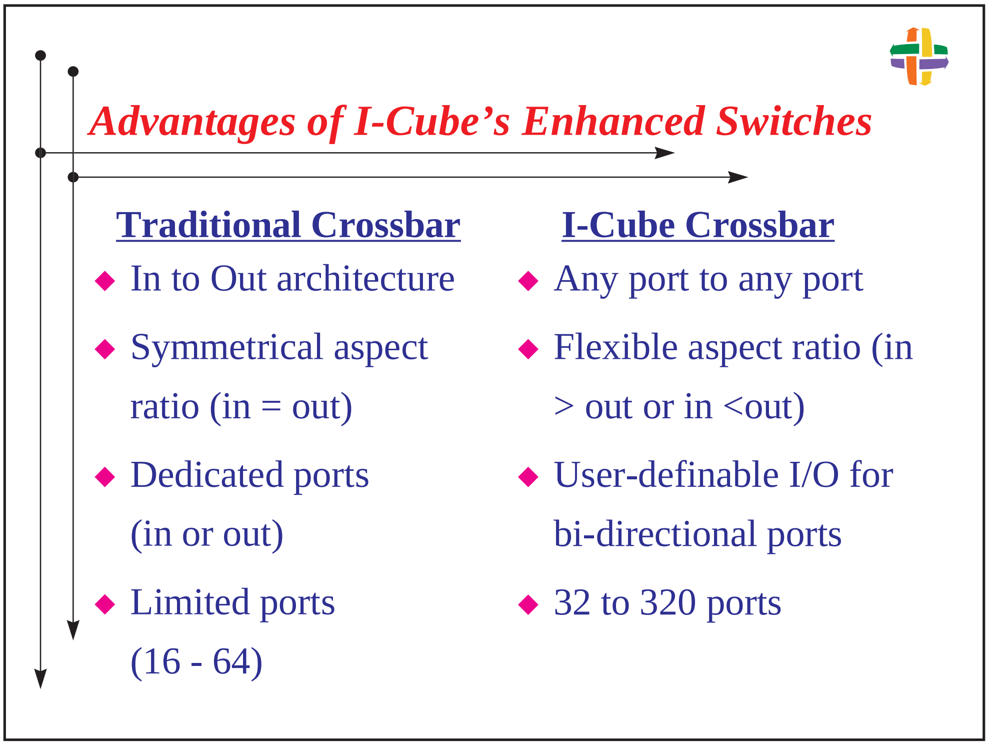
I-Cube's Crossbar Operation

1
2
Inputs
3
4
1
2
3
4
Outputs
Pass Transistor (Switch)
SRAM Cell for Switch Control
SRAM Cell = 0, Switch OPEN
SRAM Cell = 1, Switch CLOSED

The example above shows a 4 input x 4 output switch. (Note: all I/O ports can be bi-directional with DCS devices and are not limited to INPUT or OUTPUT)

All I-Cube devices are SRAM-based. This means a SRAM cell is used to control each pass transistor; writing a logic '0' to the cell opens the switch while a logic '1' closes the switch.

The IQ and IQX families have a single SRAM cell to control each switch/transistor, while the PSX family has dual SRAM cells to allow 'bank switching'. This feature allows the user to have one bank of cells 'in use' while he writes to the other bank (similar to a posted write). Then, by a single write instruction the user can update all the switch connections simultaneously.



Advantages of I-Cube's Enhanced Switches

Traditional Crossbar

- In to Out architecture
- Symmetrical aspect ratio (in = out)
- Dedicated ports (in or out)
- Limited ports (16 - 64)

I-Cube Crossbar

- Any port to any port
- Flexible aspect ratio (in > out or in <out)
- User-definable I/O for bi-directional ports
- 32 to 320 ports

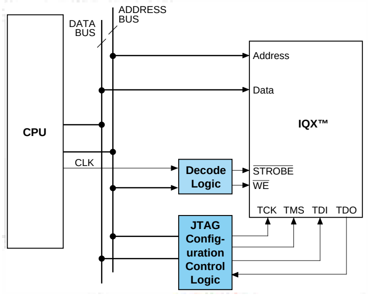
Traditional mux-based crossbars have:

- •Unidirectional routing; In to Out architecture limitation
- •Dedicated input and output ports (bi-directional signals use 2 ports)
- •Symmetrical aspect ratio (n x n only)

I-Cube enhanced crossbars have:

- •Dual-direction routing allowing any port to any port architecture flexibility
- •Programmable aspect ratios (ins > outs, ins = outs, ins < outs)
- •Programmable I/O Ports allow bi-directional signals (save I/O's)

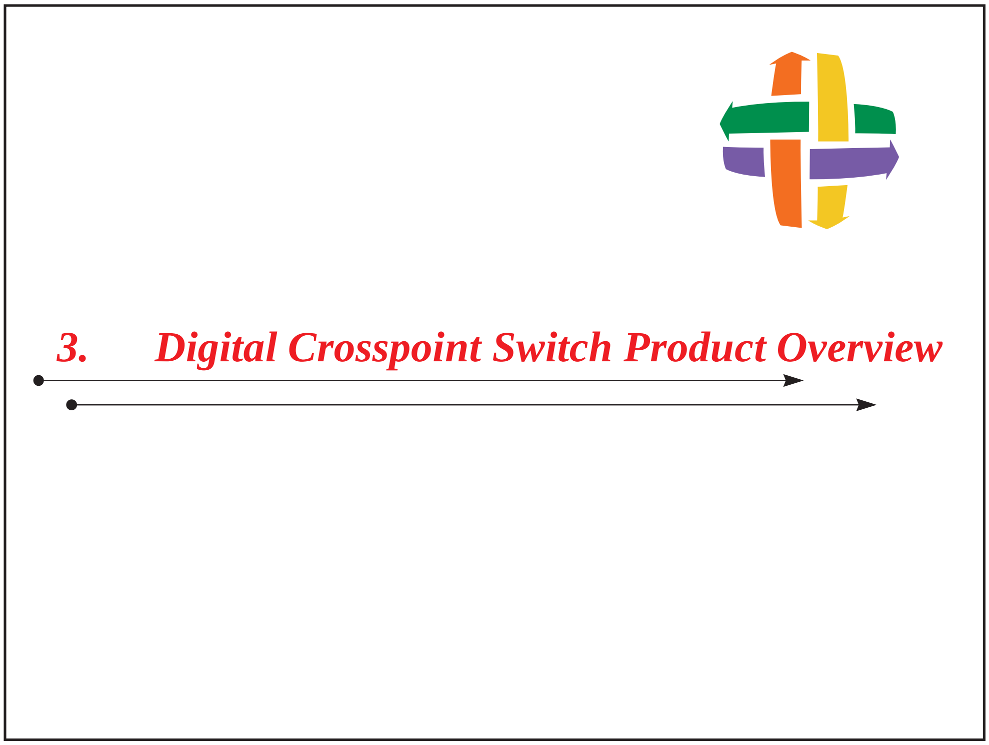
Switch Control / Interface



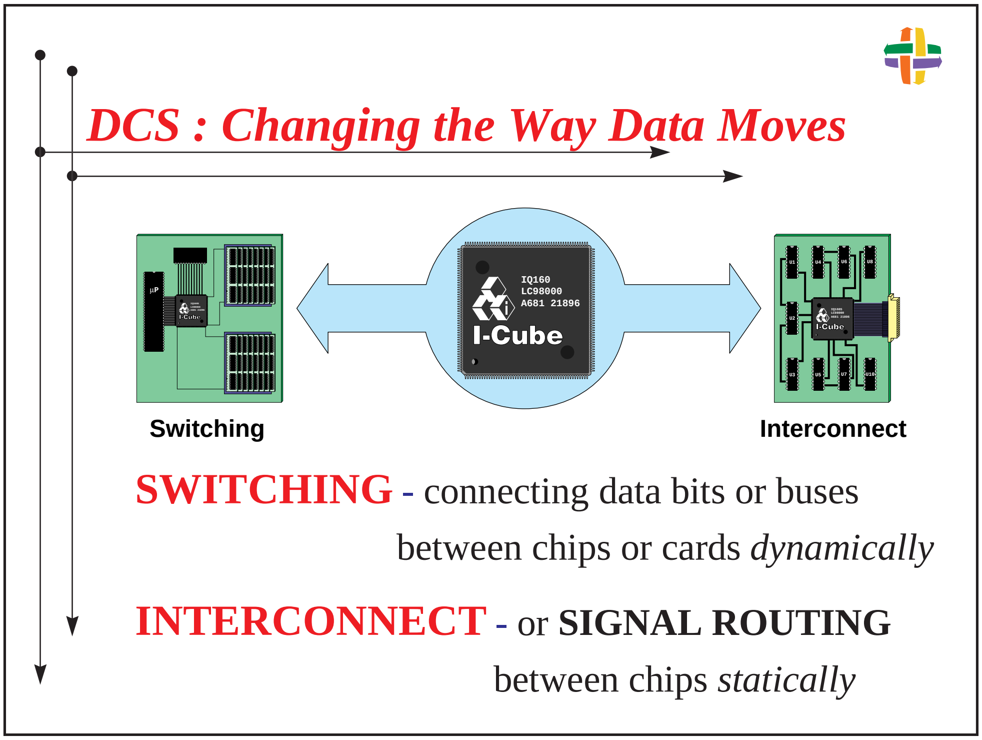
- Serial Interface:
  - *(JTAG)*
  - Configure device & make initial connections
  - Ideal for static changes
- Parallel Interface:
  - *(RapidConfigure)*
  - Gives Users Direct Control of Switch Matrix Resources
  - Ideal For Dynamic and Incremental switching
  - Change (Make or Break) Connections in nanoseconds
  - Can be Driven by CPU or State Machine

I-Cube’s Switch devices use either a serial interface called JTAG or a parallel interface called RapidConfigure. These interfaces are used to program the SRAM cells which control I/O block configuration as well as the switch matrix. The parallel interface is used to directly address the SRAM cells in the switch matrix and makes it possible to change connections very quickly - in a single “write” cycle, which could be as little as 20 ns. On the IQX devices, this interface can also be used for I/O Port configuration. The ability to use this interface for both I/O and switch configuration may eliminate the need for the JTAG interface altogether in certain applications.

The interface uses an address bus, data bus and some control signals and is similar to the CPU to memory interface found in a typical microprocessor based system. Using the address bus you select a switch matrix cell (or multiple cells on some devices) and the data bus contains the data to be written to the cell(s). The control signals are Write Enable (or chip select) and Write Strobe. The width of the address and data bus is different for the different families and devices within a family. RapidConfigure is a write-only interface. It cannot be used to read back the configuration status of the switch or I/O Ports.



3. *Digital Crosspoint Switch Product Overview*



DCS : Changing the Way Data Moves

SWITCHING - connecting data bits or buses between chips or cards *dynamically*

INTERCONNECT - or **SIGNAL ROUTING** between chips *statically*

The same crosspoint switch devices performs two different functions - and the functions are different only in the way designers understand them. We call them switching and interconnect.

Switching has a notation of being dynamic in some sense, where connections are being changed on-the-fly in real time in an embedded system.

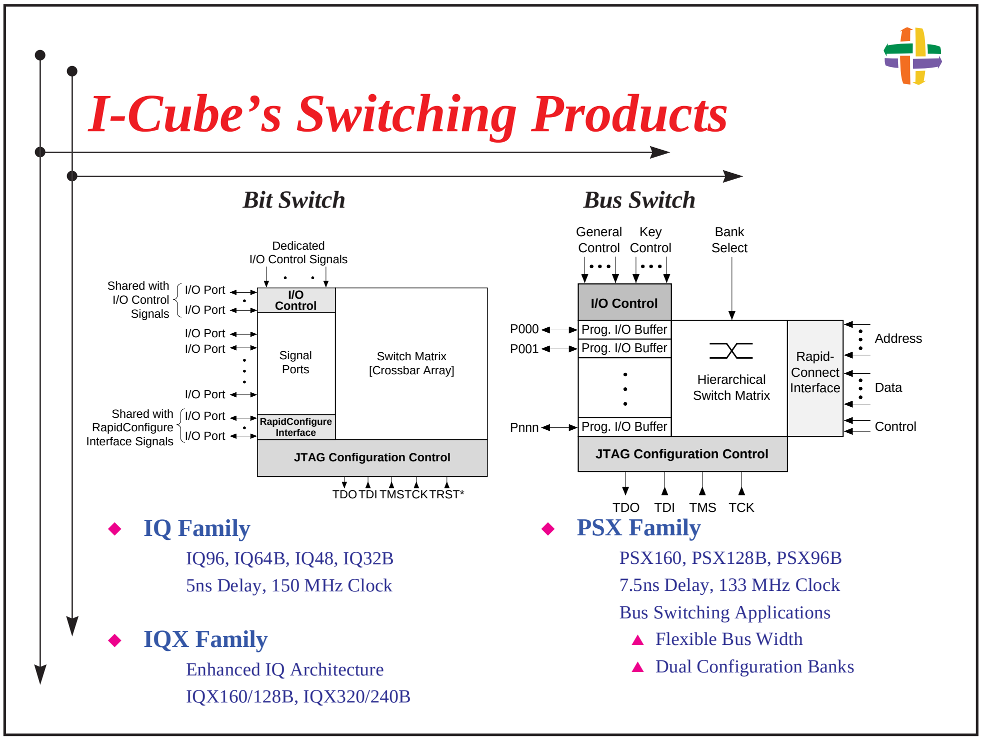
Interconnect, on the other hand, is static or pseudo-static in nature. In interconnect applications, the connections remain unchanged or are changed rather in frequently (once every few seconds/minutes/hour).

Switching

- Switched Backplanes
- Data Aggregation
- Data Muxing
- Packet Switching

Switching

- Backplane Routing
- Data Scrambling
- FPGA Interconnect



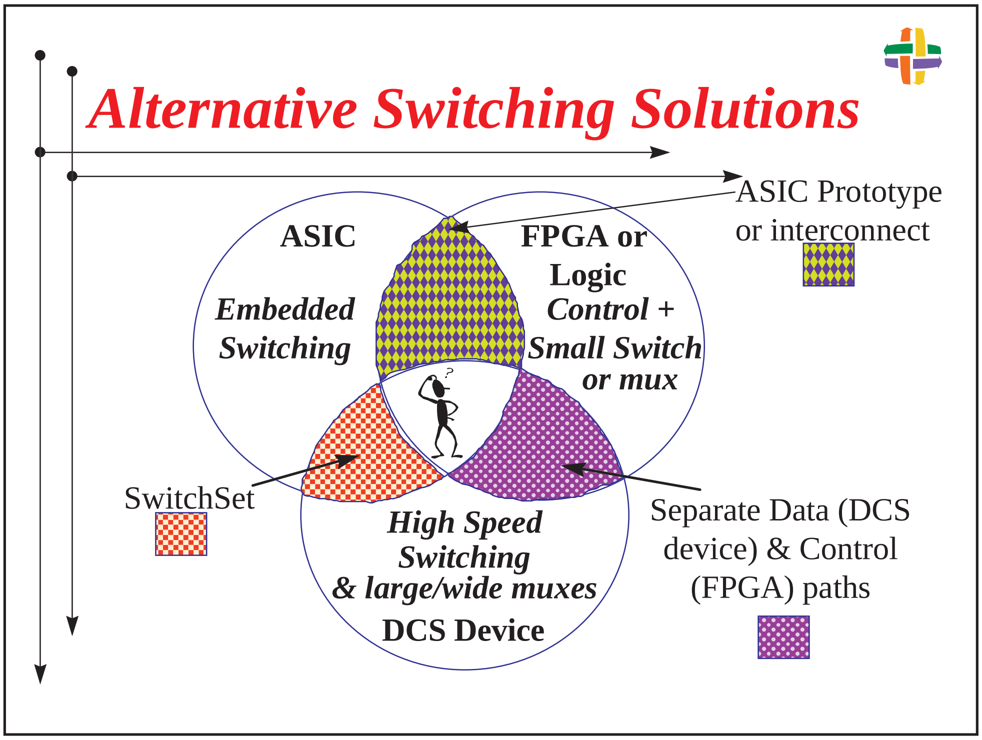
I-Cube's Switching Products

- **IQ Family**
  - IQ96, IQ64B, IQ48, IQ32B
  - 5ns Delay, 150 MHz Clock
- **IQX Family**
  - Enhanced IQ Architecture
  - IQX160/128B, IQX320/240B
- **PSX Family**
  - PSX160, PSX128B, PSX96B
  - 7.5ns Delay, 133 MHz Clock
  - Bus Switching Applications
    - Flexible Bus Width
    - Dual Configuration Banks

There are three product families in the DCS product line. Two of them, called the **IQ** and **IQX** are optimized for "**bit" switching applications** while the third family, **PSX**, is designed specifically for **"bus" switching** or for switching wider data paths.

All switch devices are described by the number of user I/Os available on the device. For example the IQX320, our largest device has 320 user I/Os, PSX160 has 160 user I/Os etc. A single die is used to support two device sizes. For example the 320 I/O die forms the IQX320 as well as the IQX240B. The 'B' suffix is used to denote that only a subset of the I/O's (in this case 240) are bonded out to pins.

The first devices introduced were the **IQ** devices. These devices are targeted at dynamic bit switching (i.e., switching of serial data streams) applications. There are eight devices in the IQ family covering a large I/O range. The devices support very high clock speeds and data rates. **IQX** devices offer enhanced features over the IQ products whilst remaining pin compatible. There are four devices in the IQX family supporting from 128 to 320 I/O's. **PSX** devices are specifically designed for "bus" switching applications and offer features such as dual configuration banks. Currently there are three devices in the PSX family.



Alternative Switching Solutions

ASIC

Embedded Switching

FPGA or Logic

Control + Small Switch or mux

ASIC Prototype or interconnect

SwitchSet

High Speed Switching & large/wide muxes

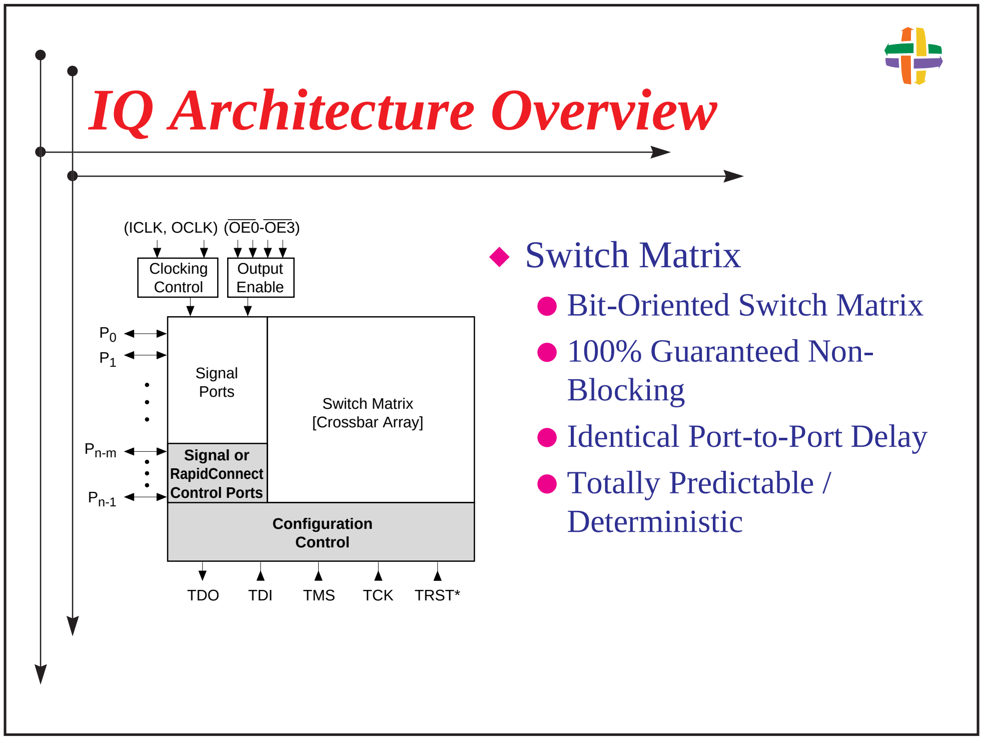
DCS Device

Separate Data (DCS device) & Control (FPGA) paths

A question that is often asked is what kind of devices I-Cube's switches are competing with. The simple answer is there are **no direct competitors**. However, there are alternatives to using DCS devices as shown above. Some functions performed by DCS devices can be done using **discrete logic** (muxes, transceivers or quickswitches). However, this approach is limited to designs requiring a small number of I/O's. I-Cubes switches offer higher flexibility with far less PCB area. Another alternative is an **FPGA**, but this option is usually limited to applications when data rates are not too high. Compared to FPGAs, I-Cube switch devices offer higher performance, have uniform predictable delays, and greater flexibility with guaranteed connectivity (no place & route required).

There are **specialized switches** such as GaAs or bipolar (ECL) switches. These are constrained to small switch sizes (16 x 16 typically) meaning you require a large number of these to perform the function that can be done with a single I-Cube device. Also when compared to I-Cube devices they are very expensive andhave a high power consumption.

Finally, a designer may implement a small crosspoint inside an **ASIC** and wrap other logic around it. However, such designs have the overhead of risk, NRE and engineering resources required.



IQ Architecture Overview

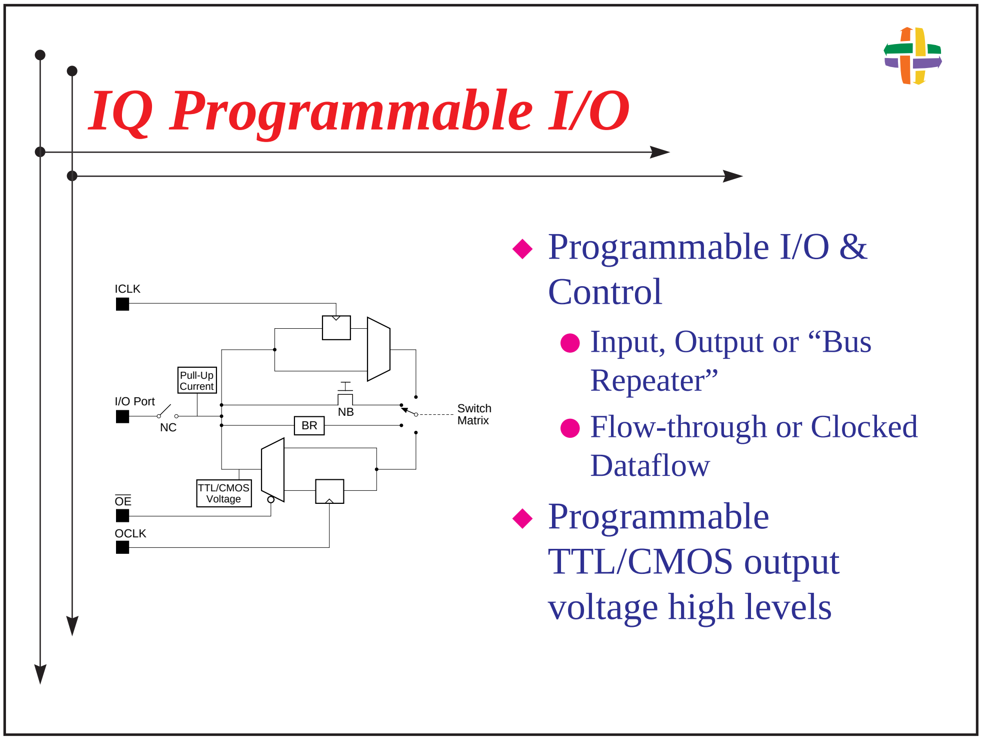
- Switch Matrix
  - Bit-Oriented Switch Matrix
  - 100% Guaranteed Non-Blocking
  - Identical Port-to-Port Delay
  - Totally Predictable / Deterministic

The IQ family of bit switching devices was the first family introduced by I-Cube. There are 4 devices based on 0.6µm CMOS process in the family.

The switch matrix in the IQ family is a bit oriented and 100% non-blocking.

The I/O ports are individually programmable for the direction and data flow. There are two global clock lines, one can be used to clock data into the device while the other is to clock data out of the device.

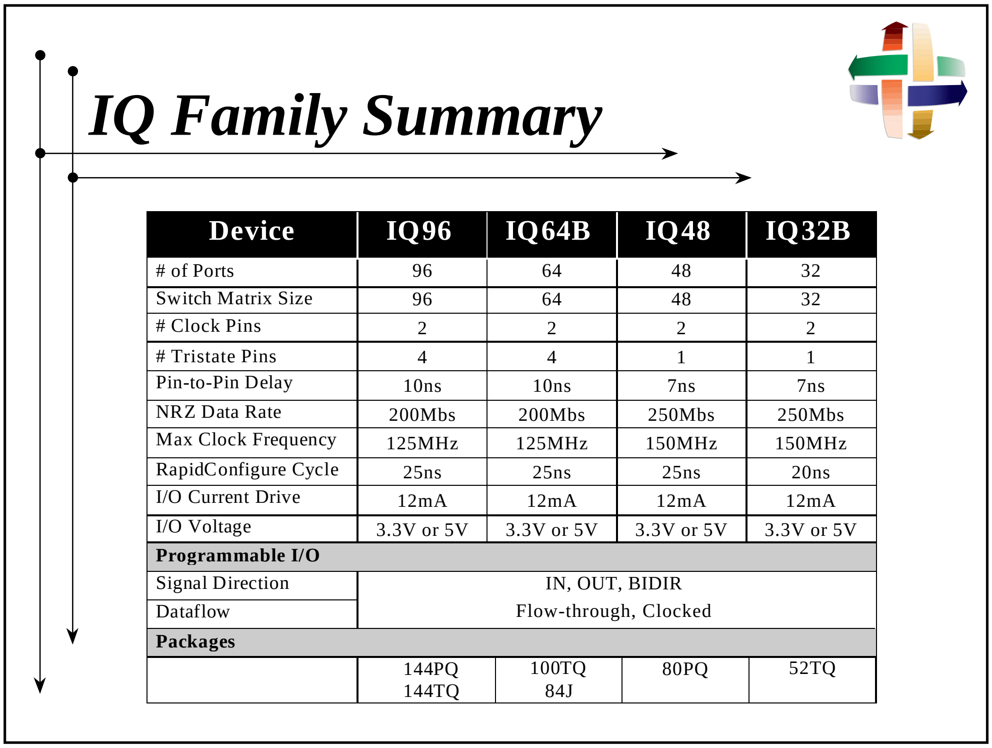
IQ devices **must** use the JTAG interface for initial switch and I/O configuration. For the IQ devices the parallel switch interface is known as “RapidConnect.”



IQ Programmable I/O

- Programmable I/O & Control
  - Input, Output or “Bus Repeater”
  - Flow-through or Clocked Dataflow
- Programmable TTL/CMOS output voltage high levels

The I/O ports are individually programmable for the direction and data flow. There are two global clock lines, one can be used to clock data into the device while the other is to clock data out of the device.



IQ Family Summary

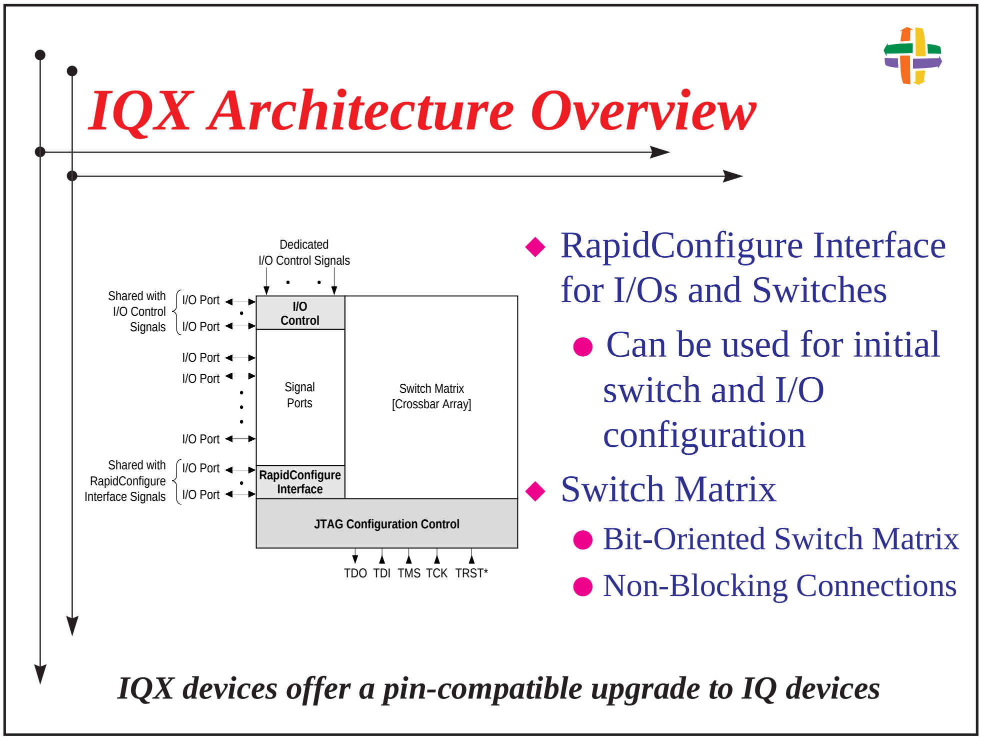
Device	IQ96	IQ64B	IQ48	IQ32B
# of Ports	96	64	48	32
Switch Matrix Size	96	64	48	32
# Clock Pins	2	2	2	2
# Tristate Pins	4	4	1	1
Pin-to-Pin Delay	10ns	10ns	7ns	7ns
NRZ Data Rate	200Mbs	200Mbs	250Mbs	250Mbs
Max Clock Frequency	125MHz	125MHz	150MHz	150MHz
RapidConfigure Cycle	25ns	25ns	25ns	20ns
I/O Current Drive	12mA	12mA	12mA	12mA
I/O Voltage	3.3V or 5V	3.3V or 5V	3.3V or 5V	3.3V or 5V
Programmable I/O				
Signal Direction	IN, OUT, BIDIR			
Dataflow	Flow-through, Clocked			
Packages				
	144PQ 144TQ	100TQ 84J	80PQ	52TQ

The table above summarizes the IQ family devices and features.

Note that the IQ32B-IQ96 devices have separate supply voltages for core and I/O buffers allowing 3.3V I/Os to be supported.

Packages:

J=PLCC, PQ=PQFP, TQ=TQFP



IQX Architecture Overview

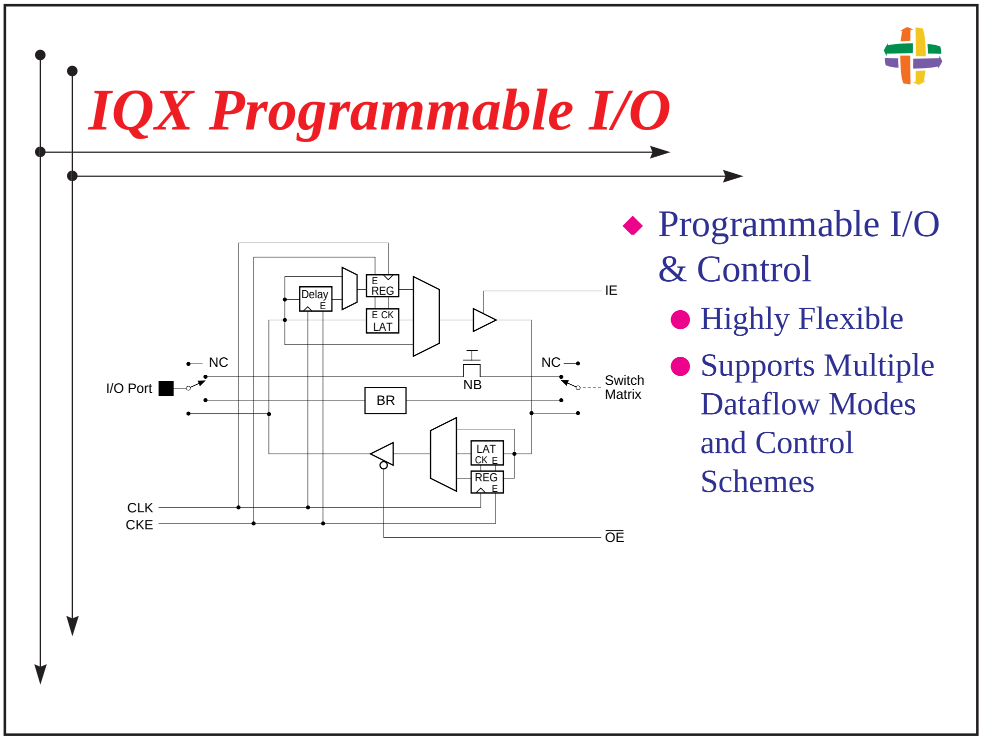
- RapidConfigure Interface for I/Os and Switches
  - Can be used for initial switch and I/O configuration
- Switch Matrix
  - Bit-Oriented Switch Matrix
  - Non-Blocking Connections

IQX devices offer a pin-compatible upgrade to IQ devices

IQX is the enhanced bit switching family based on 0.6 µm CMOS process. Four devices are available in the family. The IQX160 and IQX128B and IQX320 and IQX240B.

The IQX family uses a bit-oriented switch matrix and programmable I/O ports that are very similar to the I/O ports on the PSX. A small number of I/O ports on the device are dual purpose. They can either be used for signal I/O or for some of the special signals like RapidConfigure signals and I/O control signals.

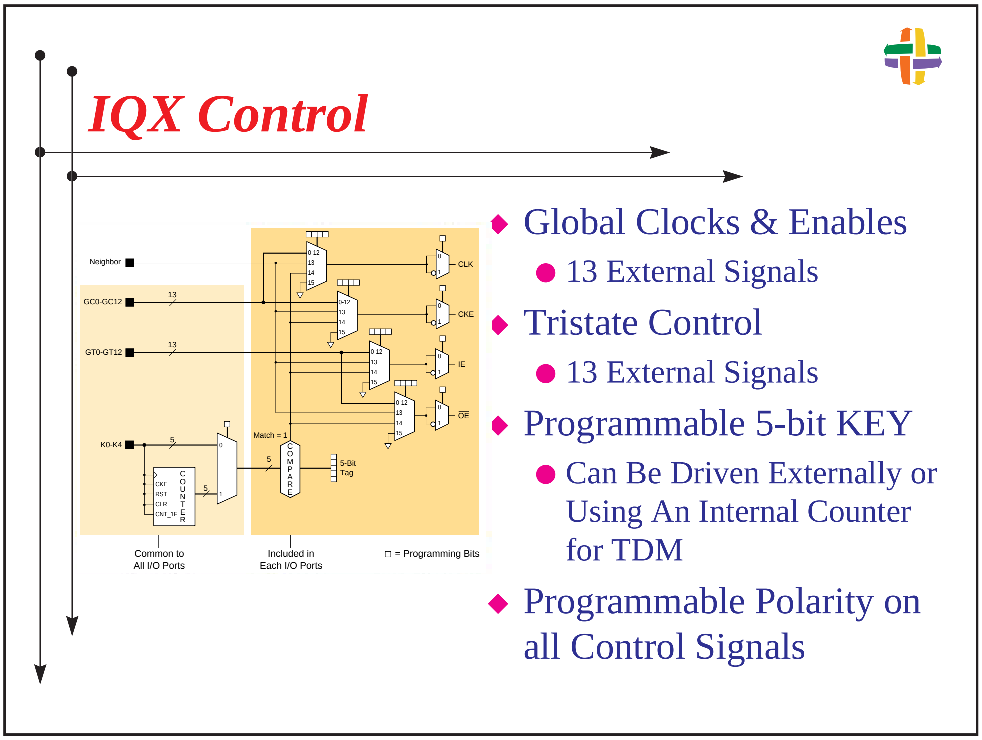
With the IQX family, the RapidConfigure interface may be used for initial switch and I/O configuration, potentially avoiding the need to use the JTAG interface.



IQX Programmable I/O

- Programmable I/O & Control
  - Highly Flexible
  - Supports Multiple Dataflow Modes and Control Schemes

The I/O Ports can be programmed for direction, data flow, and tristatablity. All I/O ports have an identical structure as shown in the bottom figure. There are four control signals, Clock, Clock Enable, Input Enable and Output Enable that control the behavior of each I/O Port.

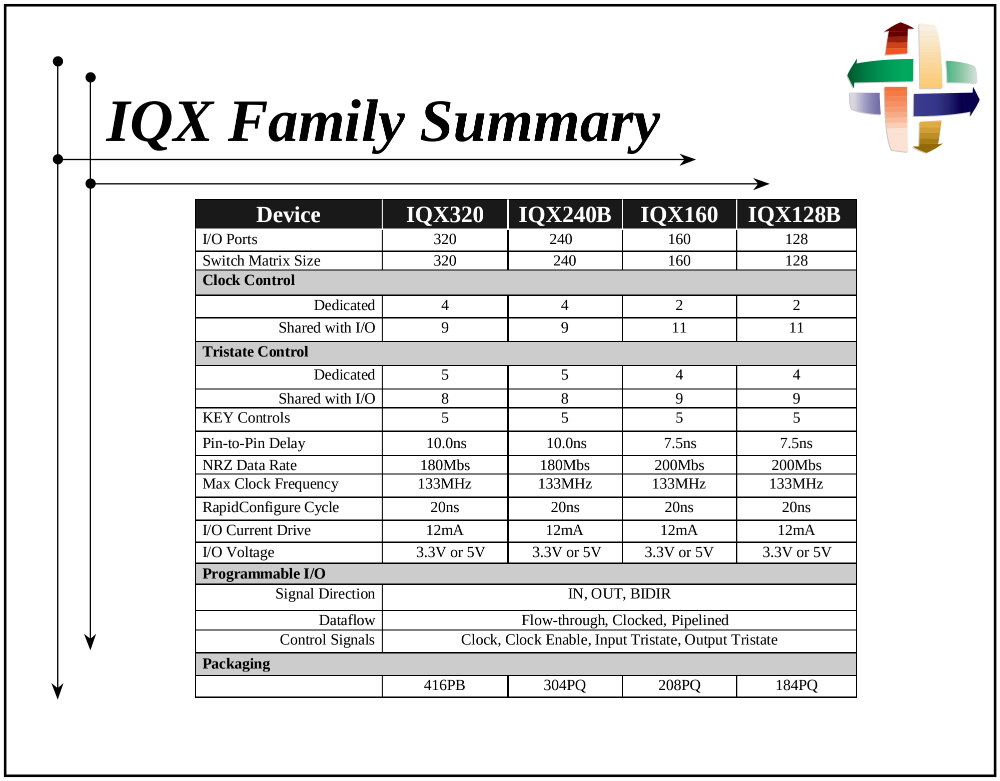


IQX Control

- Global Clocks & Enables
  - 13 External Signals
- Tristate Control
  - 13 External Signals
- Programmable 5-bit KEY
  - Can Be Driven Externally or Using An Internal Counter for TDM
- Programmable Polarity on all Control Signals

The source for the control signals is a large pool of pins - some of which are dedicated while some are shared with signal I/Os - that are used to carry the control signals into the device. Each I/O Port can then be individually and independently programmed to use any one or more of these control signals, as shown in the figure on the top. Furthermore, the polarity of these control signals can also be programmed.

In addition to the dedicated control signal sources, a 5-pin “KEY” is used to generate mutually exclusive control signals. Each I/O Port contains a programmable 5-bit tag that is continuously compared against the data on the 5-pin KEY port. A match signal is generated when the key value matches with the programmed tag. This match signal can be used as one additional source for the control signal. The ability to tristate in the input direction allows you to create internal bus structures on the device. These on-chip bus structures can be operated at very high speeds. For example, a combination of this capability and the key feature allows you to build a 32 slot or lower TDM highway that can be operated at 80 MHz clock rates.



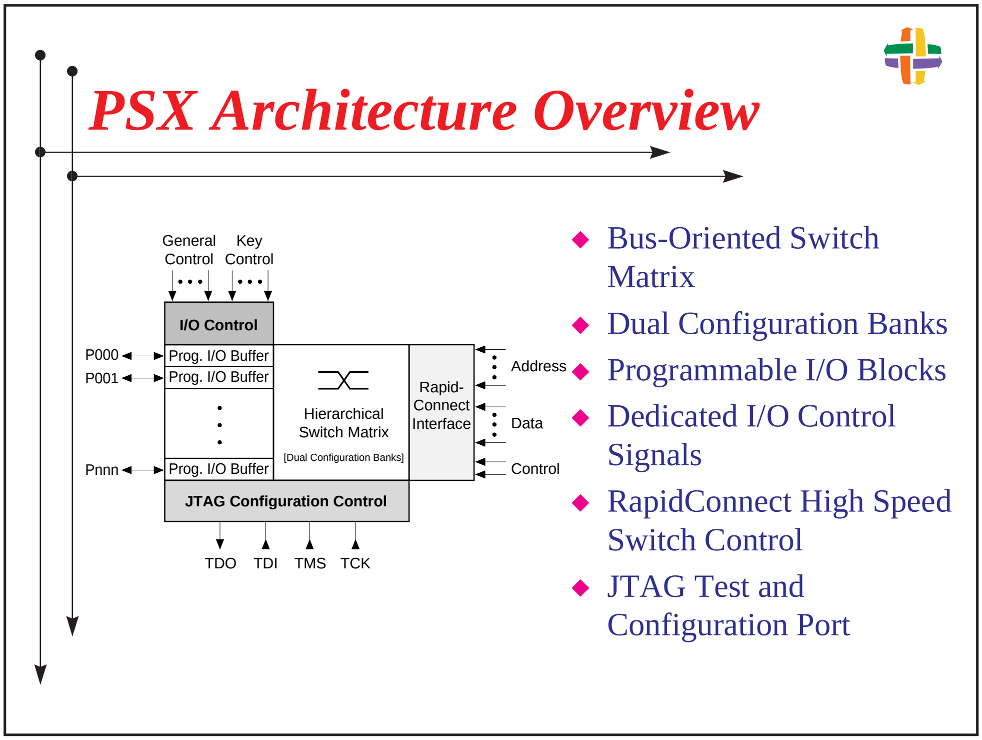
IQX Family Summary

Device	IQX320	IQX240B	IQX160	IQX128B
I/O Ports	320	240	160	128
Switch Matrix Size	320	240	160	128
Clock Control				
Dedicated	4	4	2	2
Shared with I/O	9	9	11	11
Tristate Control				
Dedicated	5	5	4	4
Shared with I/O	8	8	9	9
KEY Controls	5	5	5	5
Pin-to-Pin Delay	10.0ns	10.0ns	7.5ns	7.5ns
NRZ Data Rate	180Mbs	180Mbs	200Mbs	200Mbs
Max Clock Frequency	133MHz	133MHz	133MHz	133MHz
RapidConfigure Cycle	20ns	20ns	20ns	20ns
I/O Current Drive	12mA	12mA	12mA	12mA
I/O Voltage	3.3V or 5V	3.3V or 5V	3.3V or 5V	3.3V or 5V
Programmable I/O				
Signal Direction	IN, OUT, BIDIR			
Dataflow	Flow-through, Clocked, Pipelined			
Control Signals	Clock, Clock Enable, Input Tristate, Output Tristate			
Packaging				
	416PB	304PQ	208PQ	184PQ

The table above summarizes the IQX family devices and features. The IQX320, 240B, 160 and 128B are all in production.

Packages:

PB=PBGA, PQ=PQFP



PSX Architecture Overview

- Bus-Oriented Switch Matrix
- Dual Configuration Banks
- Programmable I/O Blocks
- Dedicated I/O Control Signals
- RapidConnect High Speed Switch Control
- JTAG Test and Configuration Port

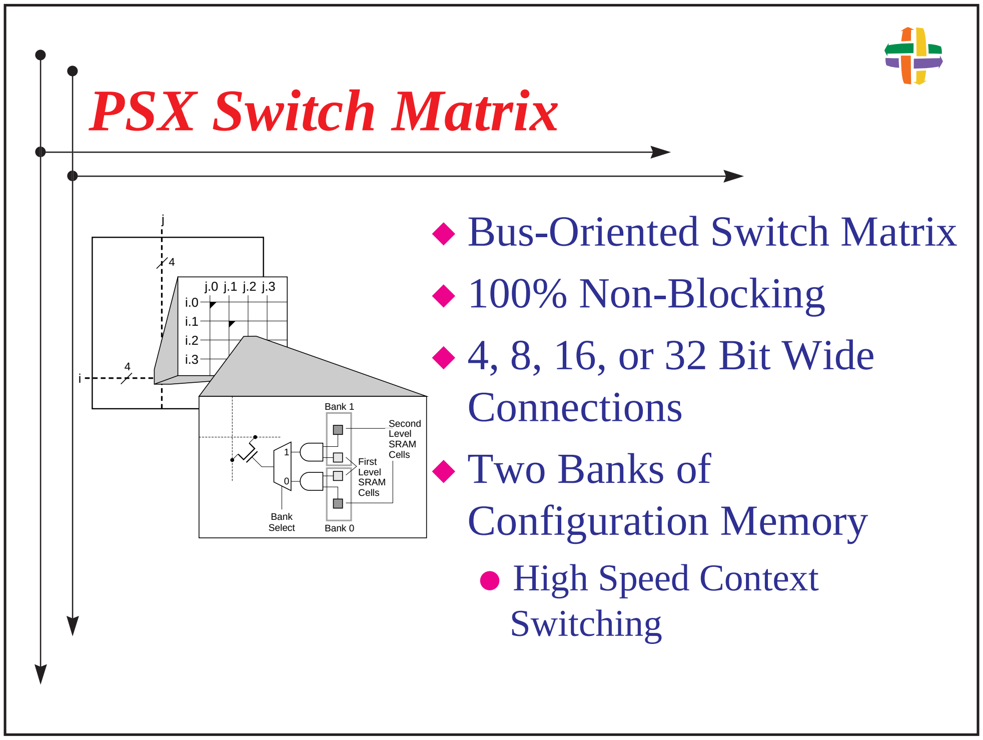
Here is a look at the PSX family. The PSX devices are targeted at dynamic, bus switching applications.

The switch matrix in the PSX is bus-oriented. That means a single SRAM cell in the switch controls the connection status of a group of I/O ports. The minimum granularity or group size is 4 or a nibble.

The I/O ports are very flexible and can be configured in many different ways. The PSX devices have a dedicated pins for RapidConfigure interface and for bringing in a large number of control signals.

Currently, there are three devices in the PSX family. They are all based on 0.6 µm CMOS process.

PSX devices **must** use the JTAG interface for initial switch and I/O configuration. For the PSX devices the parallel switch interface is known as “RapidConnect.”

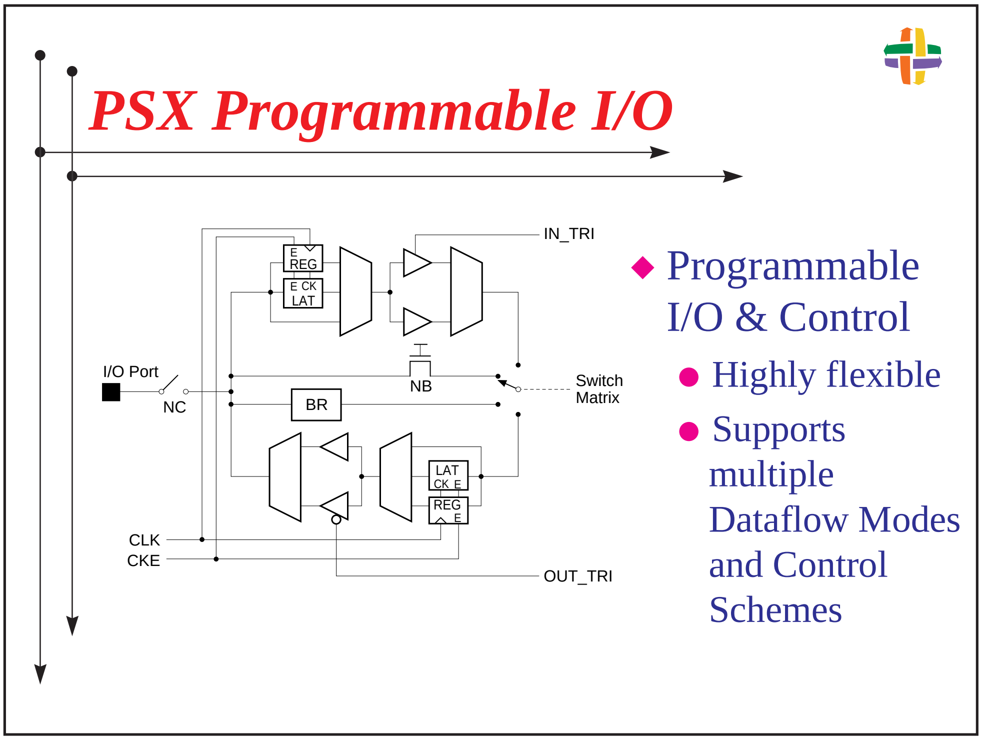


PSX Switch Matrix

- Bus-Oriented Switch Matrix
- 100% Non-Blocking
- 4, 8, 16, or 32 Bit Wide Connections
- Two Banks of Configuration Memory
  - High Speed Context Switching

The PSX switch matrix is bus oriented, with a minimum bus size of 4-bits or a nibble. In addition, the switch is constructed in a hierarchical fashion. This structure allows you to build wider busses that are 2, 4 or 8 times the minimum bus width, i.e., 8, 16 and 32 bits wide. This is accomplished by using two SRAM cells to control each switch. We call them level 1 (L1) and level 2 (L2) cells. There is one L1 cell per nibble group while there is a common L2 cell for two nibble (or a byte) group. A pass transistor (or actually a group of four pass transistors) is turned on only when its L1 and L2 cells contain a "1." When the switch is operated in the nibble mode, the contents of the L2 cells is fixed and the connection changes are accomplished by changing data stored in L1 cells. In byte, word and long word modes, the L1 contents are fixed and the changes are effected by storing the same data in a 1x1, 2x2 or 4x4 tile L2 cells.

In addition to the L1 and L2 SRAM structure, the switch matrix has dual configuration banks, called Bank A and Bank B. A pin signal called ***BankSelect*** determines which bank controls the switch at any given time. Using the RapidConfigure interface you can load a new switch configuration in one bank (called the stand by bank) while the switch is operational - being controlled other bank (called the active bank). The switch matrix structure also results in identical propagation delays and between bus pairs with very tight control over skews.

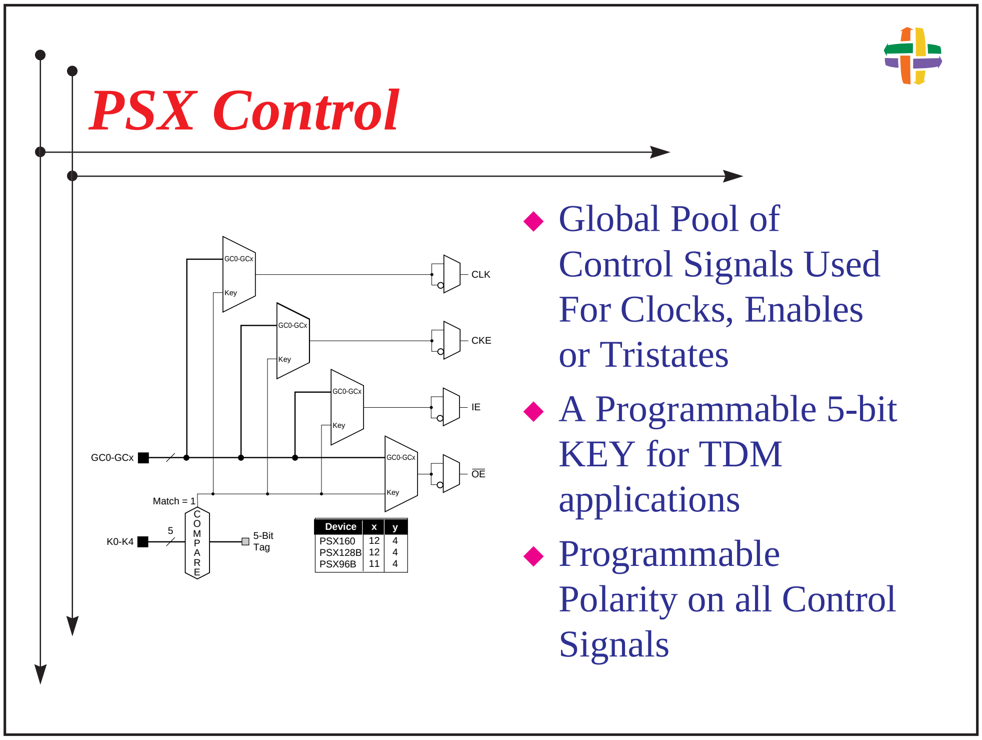


PSX Programmable I/O

◆ Programmable I/O & Control

- Highly flexible
- Supports multiple Dataflow Modes and Control Schemes

The I/O Ports can be programmed for direction, data flow, and tristatablity. All I/O ports have an identical structure as shown in the bottom figure. There are four control signals, Clock, Clock Enable, Input Enable and Output Enable that control the behavior of each I/O Port.



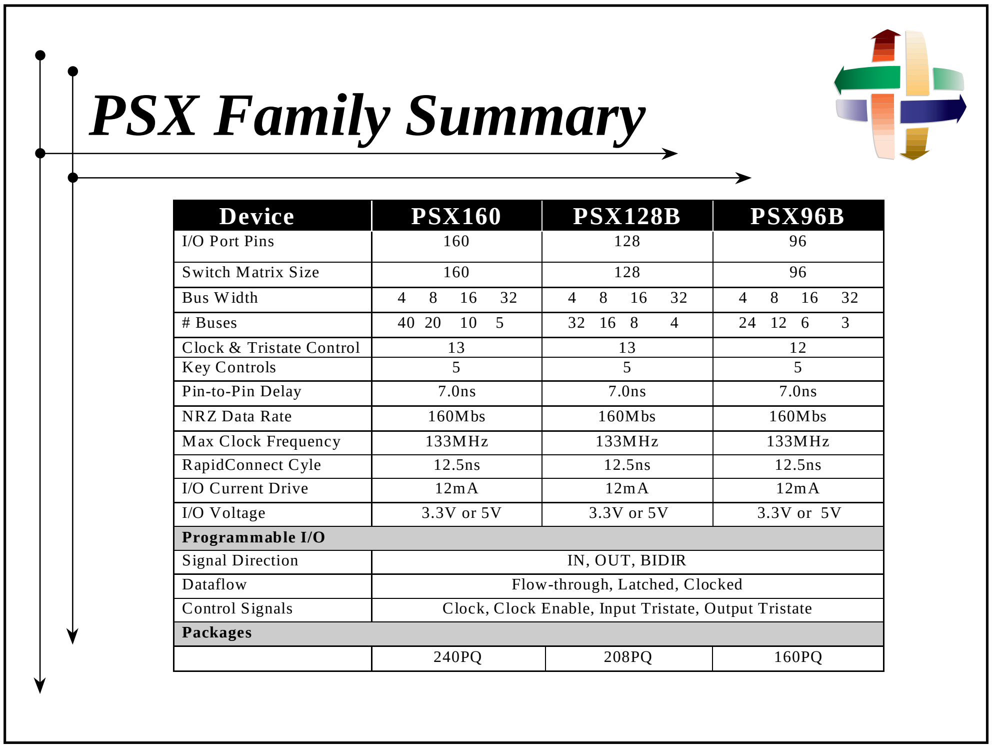
PSX Control

Device	x	y
PSX160	12	4
PSX128B	12	4
PSX96B	11	4

- Global Pool of Control Signals Used For Clocks, Enables or Tristates
- A Programmable 5-bit KEY for TDM applications
- Programmable Polarity on all Control Signals

The source for the control signals is a large pool of dedicated pins that are used to carry the control signals into the device. Each I/O Port can then be individually and independently programmed to use any one or more of these control signals, as shown in the figure on the top. Furthermore, the polarity of these control signals can also be programmed.

In addition to the dedicated control signal sources, a 5-pin “KEY” is used to generate mutually exclusive control signals. Each I/O Port contains a programmable 5-bit tag that is continuously compared against the data on the 5-pin KEY port. A match signal is generated when the key value matches with the programmed tag. This match signal can be used as one additional source for the control signal. The ability to tristate in the input direction allows you to create internal bus structures on the device. These on-chip bus structures can be operated at very high speeds. For example, a combination of this capability and the key feature allows you to build a 32 slot or lower TDM highway that can be operated at 80 MHz clock rates.



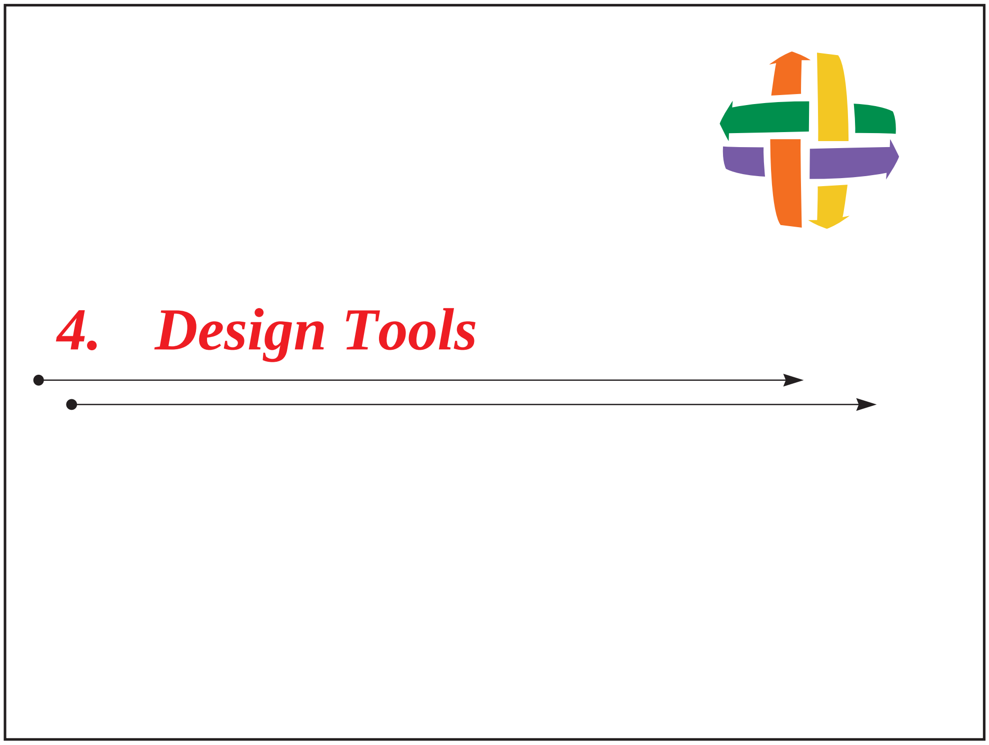
PSX Family Summary

Device	PSX160	PSX128B	PSX96B
I/O Port Pins	160	128	96
Switch Matrix Size	160	128	96
Bus Width	4 8 16 32	4 8 16 32	4 8 16 32
# Buses	40 20 10 5	32 16 8 4	24 12 6 3
Clock & Tristate Control	13	13	12
Key Controls	5	5	5
Pin-to-Pin Delay	7.0ns	7.0ns	7.0ns
NRZ Data Rate	160Mbs	160Mbs	160Mbs
Max Clock Frequency	133MHz	133MHz	133MHz
RapidConnect Cyle	12.5ns	12.5ns	12.5ns
I/O Current Drive	12mA	12mA	12mA
I/O Voltage	3.3V or 5V	3.3V or 5V	3.3V or 5V
Programmable I/O			
Signal Direction	IN, OUT, BIDIR		
Dataflow	Flow-through, Latched, Clocked		
Control Signals	Clock, Clock Enable, Input Tristate, Output Tristate		
Packages			
	240PQ	208PQ	160PQ

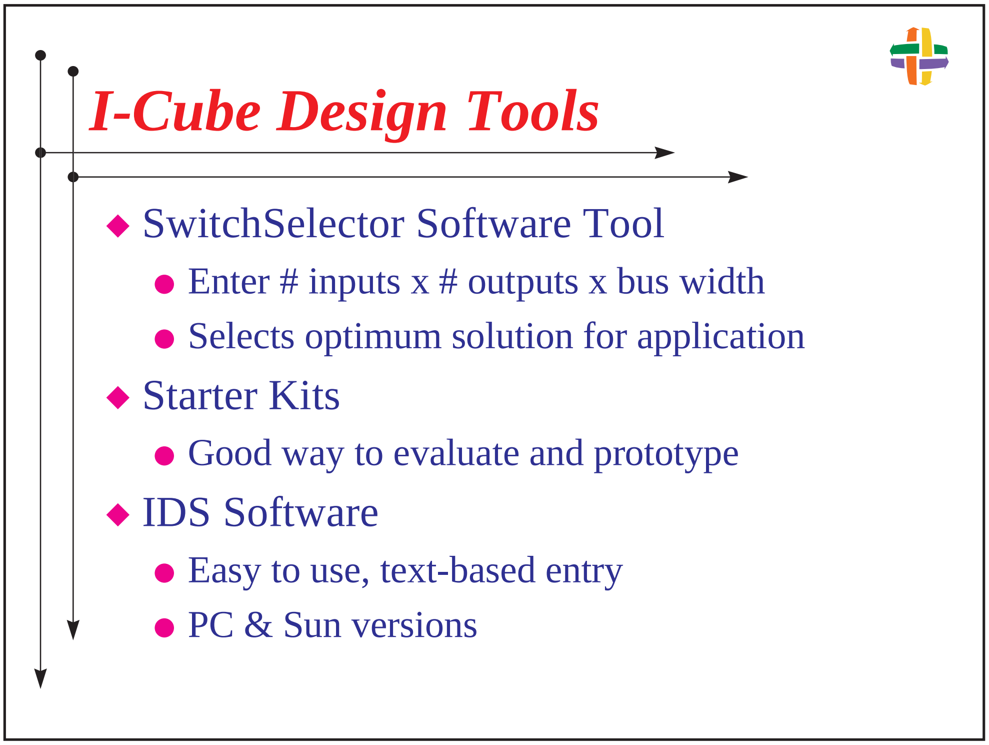
The table above summarizes the PSX family devices and features.

Packages:

PQ=PQFP



4. Design Tools



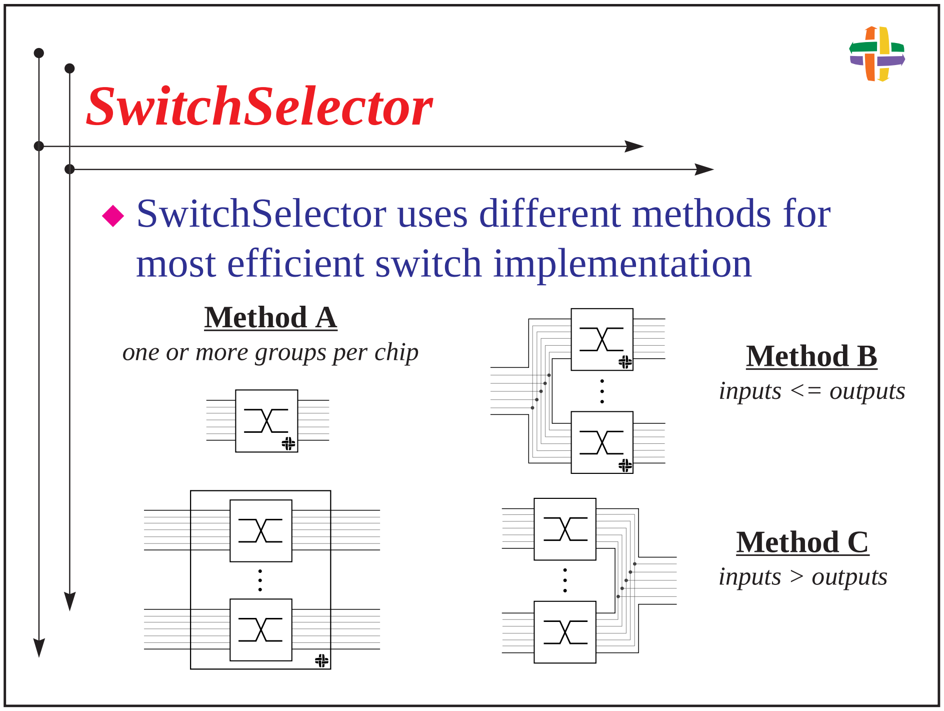
I-Cube Design Tools

- SwitchSelector Software Tool
  - Enter # inputs x # outputs x bus width
  - Selects optimum solution for application
- Starter Kits
  - Good way to evaluate and prototype
- IDS Software
  - Easy to use, text-based entry
  - PC & Sun versions

I-Cube offers a range of low cost tools to allow engineers to create designs with switch-based architectures. The SwitchSelector software tool accepts inputs such as # inputs, # outputs, bus width, and then provide guidance on how to implement the switch using one or multiple of I-Cube’s switch devices. This allows the engineer to optimize his design for minimum cost, lowest device count, smallest PCB area, etc.

As a next step, for easy, low cost and risk evaluation I-Cube offers Starter Kits which include all the software and hardware needed to build a switch.

Finally, the **I**-Cube **D**evelopment **S**ystem (IDS) software is an easy to use tool which supports ALL I-Cube devices.



SwitchSelector

- ◆ SwitchSelector uses different methods for most efficient switch implementation

Method A
one or more groups per chip

Method B
inputs <= outputs

Method C
inputs > outputs

SwitchSelector allows the user to design without having to fully understand all the features of I-Cube's crossbar switching devices. This tool will recommend how to implement large switch matrices effectively using a range of devices.

Five different methods are available, each of which has specific advantages depending upon the switch size and configuration.

Method A

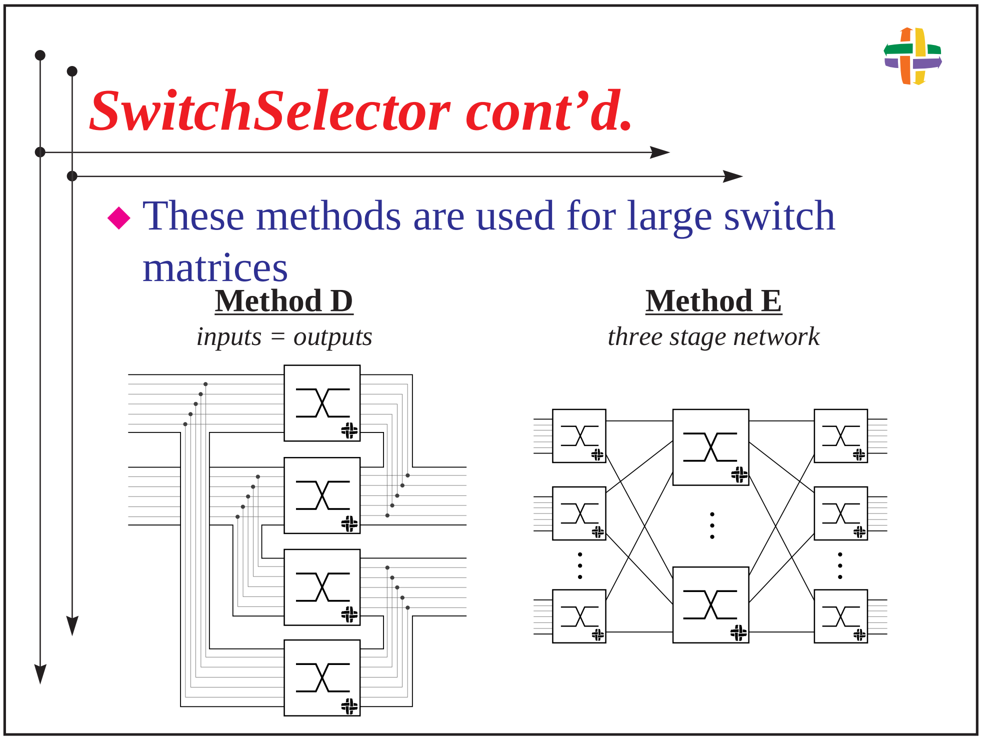
- One or more groups per chip
- (Inputs + Outputs) <= 320 for IQ and IQX
- (Inputs + Outputs) <= 40 for PSX

Method B

- Inputs <=Outputs
- Inputs are common to all chips, with the outputs split between chips

Method C

- Inputs > Outputs
- Inputs are split between chips, with the outputs common to all chips



SwitchSelector cont’d.

- These methods are used for large switch matrices

Method D

inputs = outputs

Method E

three stage network

Method D

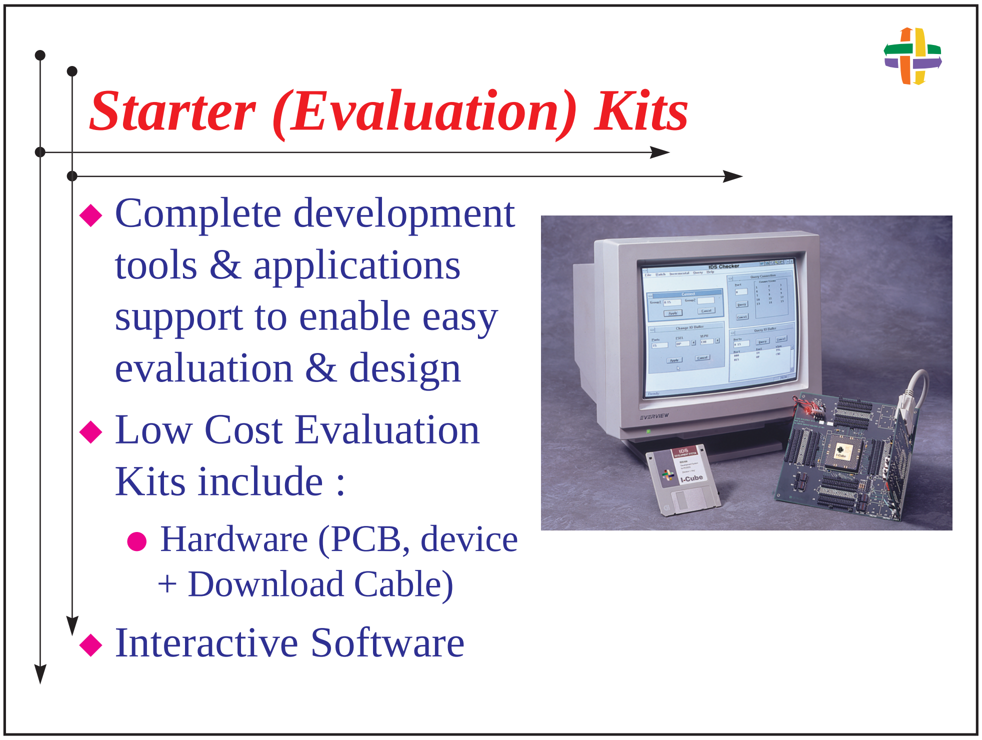
•For large ‘square’ switches

•Divides a *NxN* array into smaller *nxn* arrays

Method E

•For very large switches, where #Inputs >320 for IQX, and >40 for PSX

•A non-blocking 3 stage network which uses the same size switch for stages 1 & 3, with a bigger device for stage 2 (to ensure non-blocking behavior)



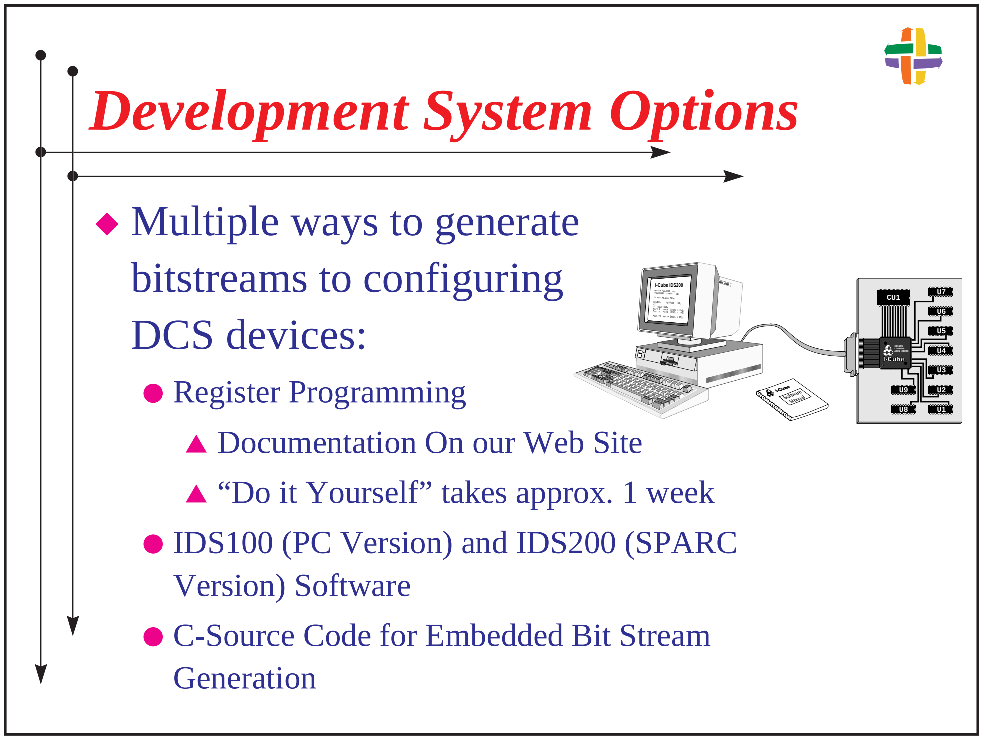
Starter (Evaluation) Kits

- Complete development tools & applications support to enable easy evaluation & design
- Low Cost Evaluation Kits include :
  - Hardware (PCB, device + Download Cable)
- Interactive Software

I-Cube's crossbar devices and switching technology will likely be 'new' to many engineers. I-Cube has an excellent, low cost starter kit which includes ALL the hardware and software required to start 'playing' with these easy to use devices.

The software runs on a PC (Windows and DOS based) and generates the bitstream required to program or 'configure' the switch. (I-Cube's switches are similar to SRAM FPGAs in needing to be configured at power on).

Starter Kits are available for IQX320, IQX160, IQ96 and PSX160. A cable is supplied to connect the PC to the circuit board via the parallel port to allow the DCS device to be configured. Additionally there is room on the PCB to breadboard a customers own circuit. As the device is SRAM based, it can be reconfigured as many times as required.



Development System Options

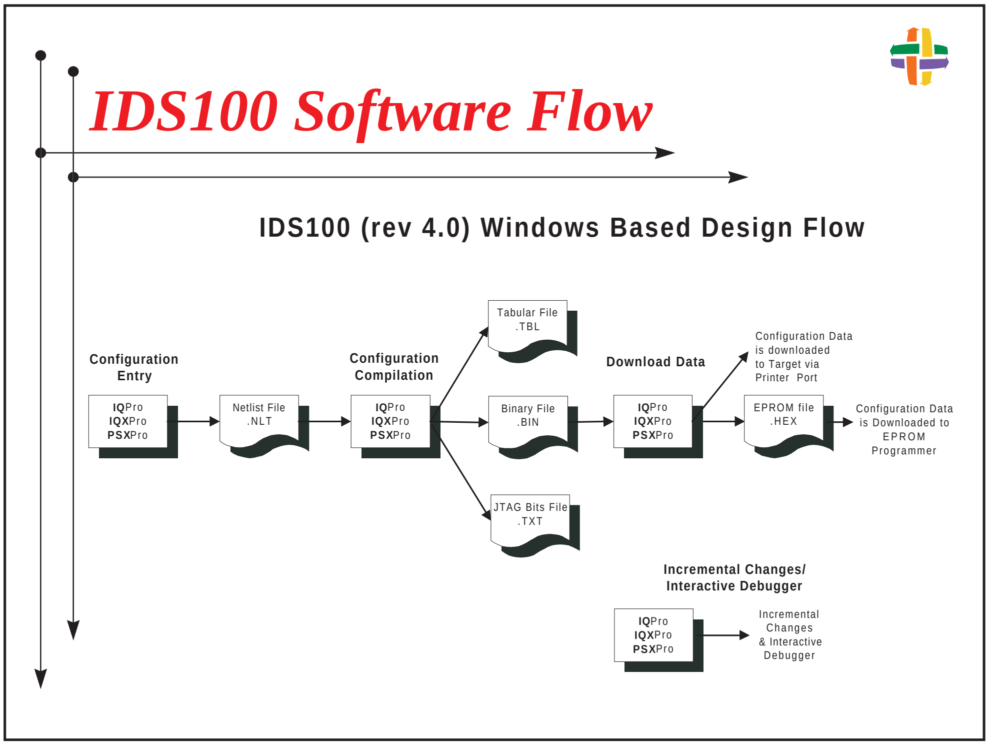
- Multiple ways to generate bitstreams to configuring DCS devices:
  - Register Programming
    - Documentation On our Web Site
    - “Do it Yourself” takes approx. 1 week
  - IDS100 (PC Version) and IDS200 (SPARC Version) Software
  - C-Source Code for Embedded Bit Stream Generation

CU1
U7
U6
U5
U4
U3
U2
U1
U9
U8

Now the engineer has decided to design with an I-Cube switch, we can simplify the design process: There are several different ways to generate the bit stream for configuring I-Cube’s switch devices. The optimal choice will be different for different applications.

The first option is called off-line bit stream generation. This method is suited for applications requiring static or pseudo-static interconnect or routing. Under this option, the necessary bit stream(s) are generated ahead of time. They can then be stored in a non-volatile memory like flash or EPROM or they could be downloaded from a host such as a PC.

We offer two products that allow the user to generate the bit stream - PC/Windows product called the **IDS100** and SparcStation version called the **IDS200**. They both have the same price.



IDS100 Software Flow

IDS100 (rev 4.0) Windows Based Design Flow

Configuration Entry

IQPro
IQXPro
PSXPro

Netlist File
.NLT

Configuration Compilation

IQPro
IQXPro
PSXPro

Tabular File
.TBL

Binary File
.BIN

JTAG Bits File
.TXT

Download Data

IQPro
IQXPro
PSXPro

Configuration Data is downloaded to Target via Printer Port

EPROM file
.HEX

Configuration Data is Downloaded to EPROM Programmer

Incremental Changes/ Interactive Debugger

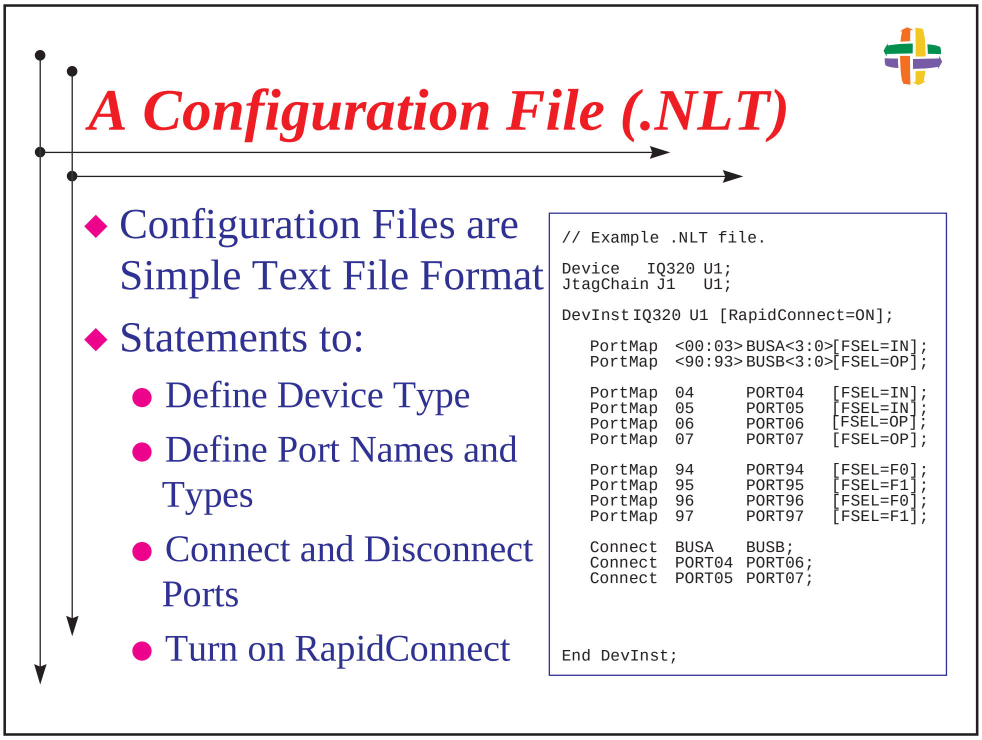
IQPro
IQXPro
PSXPro

Incremental Changes & Interactive Debugger

This shows the software flow for WINDOWS based tools, from the text based configuration file (.NLT), which is then compiled and gives multiple outputs, to the downloading of the bitstream and the interactive debugging.

IQPro/IQXPro/PSXPro provide a bitstream compiler, downloader, interactive debugger and Hex file generation for the relevant device family.

A DOS based design flow is also supported.

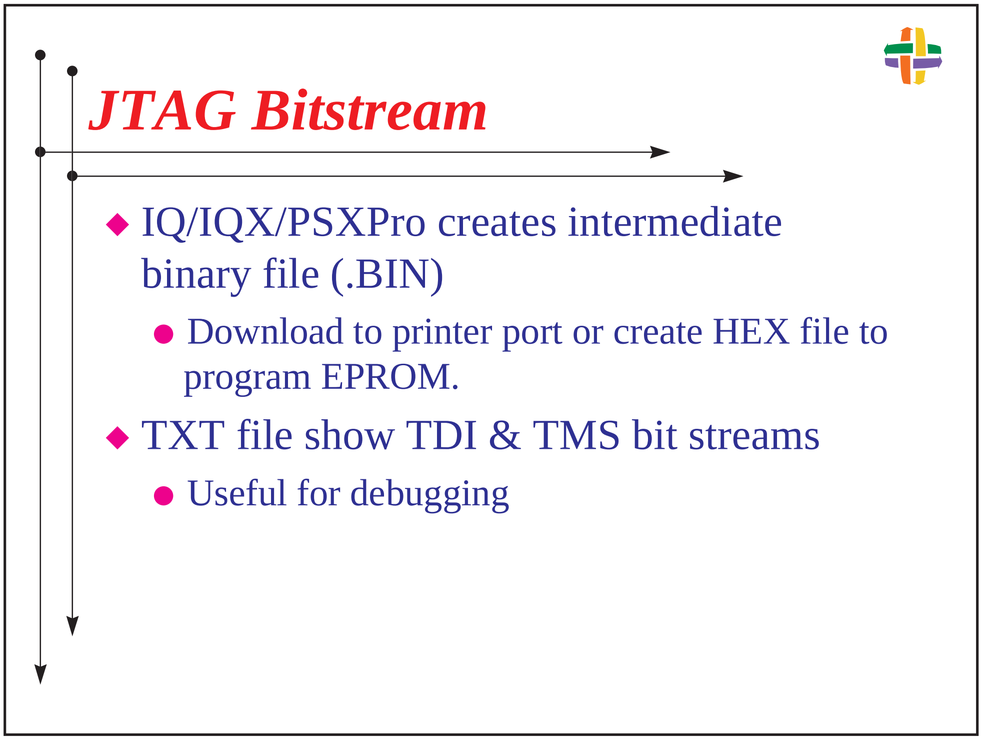


A Configuration File (.NLT)

- Configuration Files are Simple Text File Format
- Statements to:
  - Define Device Type
  - Define Port Names and Types
  - Connect and Disconnect Ports
  - Turn on RapidConnect

```
// Example .NLT file.

Device    IQ320 U1;
JtagChain J1    U1;

DevInst IQ320 U1 [RapidConnect=ON];

   PortMap  <00:03> BUSA<3:0>[FSEL=IN];
   PortMap  <90:93> BUSB<3:0>[FSEL=OP];

   PortMap  04      PORT04   [FSEL=IN];
   PortMap  05      PORT05   [FSEL=IN];
   PortMap  06      PORT06   [FSEL=OP];
   PortMap  07      PORT07   [FSEL=OP];

   PortMap  94      PORT94   [FSEL=F0];
   PortMap  95      PORT95   [FSEL=F1];
   PortMap  96      PORT96   [FSEL=F0];
   PortMap  97      PORT97   [FSEL=F1];

   Connect  BUSA    BUSB;
   Connect  PORT04  PORT06;
   Connect  PORT05  PORT07;


End DevInst;
```

This is an example of a configuration file used by the IDS100 or IDS200 software. It is used to describe the configuration of the I/O ports and connections through the switch matrix.



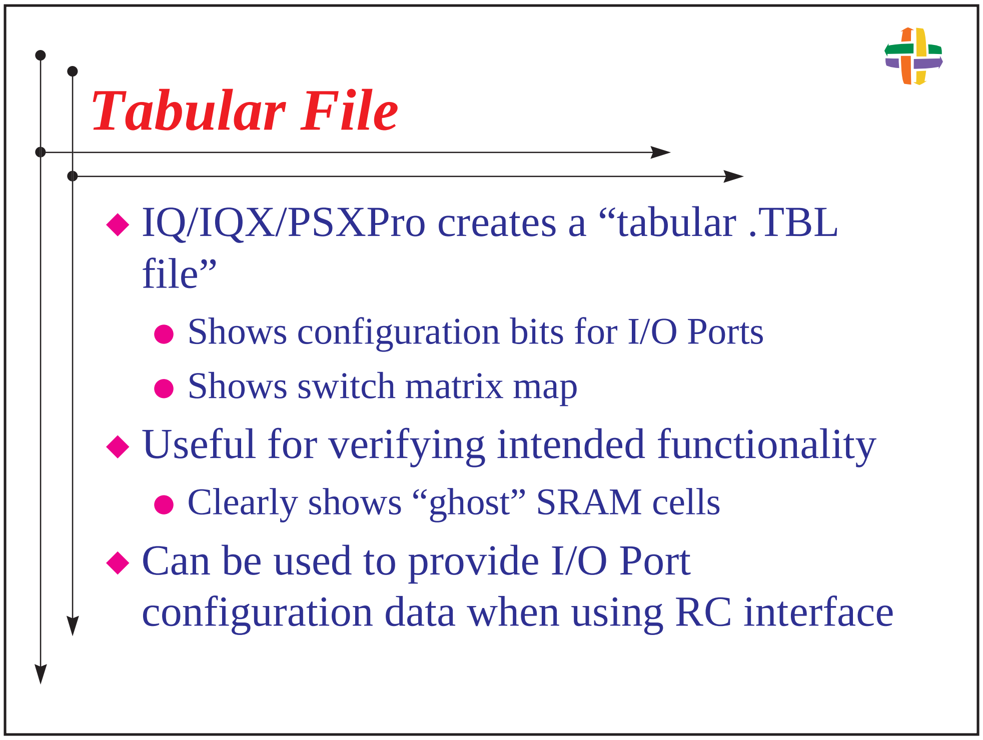
JTAG Bitstream

- ◆ IQ/IQX/PSXPro creates intermediate binary file (.BIN)
  - ● Download to printer port or create HEX file to program EPROM.
- ◆ TXT file show TDI & TMS bit streams
  - ● Useful for debugging

All of I-Cube’s switch devices feature a TAP (Test Access Port) compliant to the IEEE 1149.1 (JTAG) Standard. I-Cube use extended JTAG-TAP capabilities for first loading the initial configuration and subsequently for configuration changes.

I-Cube devices need two bit stream patterns, one applied to TMS (Test Mode Select) and another applied to TDI (Test Data Input).

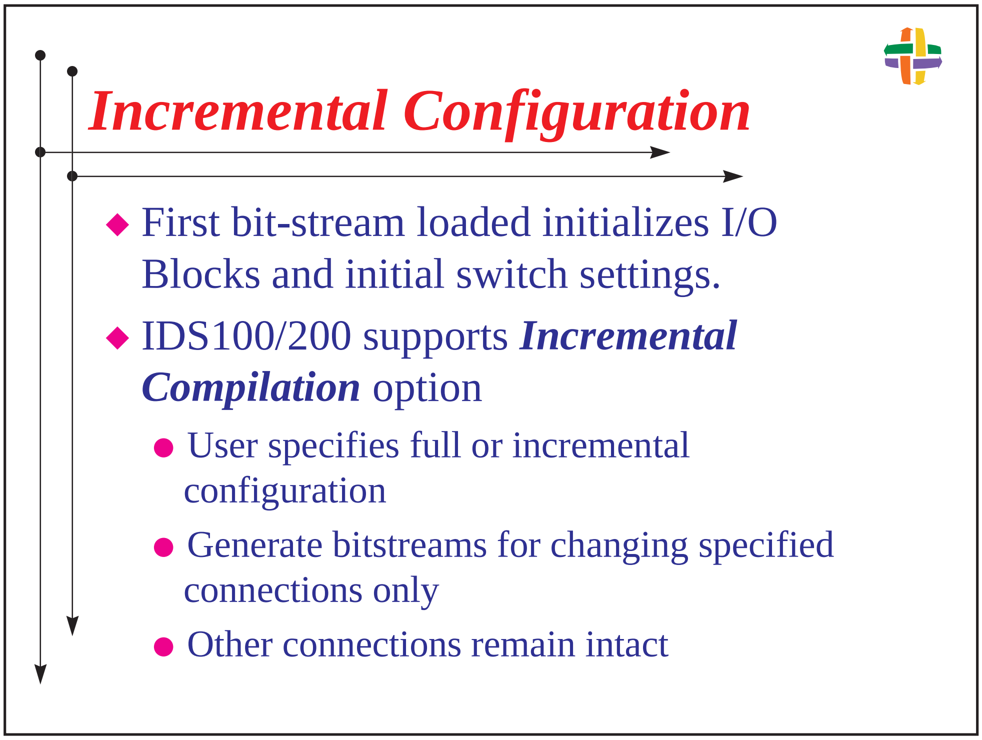
The IDS software generates the configuration-information (TDI) bitstream and a second accompanying bitstream (TMS) which is used to control the TAP state machine in order to properly sequence loading.



Tabular File

- IQ/IQX/PSXPro creates a “tabular .TBL file”
  - Shows configuration bits for I/O Ports
  - Shows switch matrix map
- Useful for verifying intended functionality
  - Clearly shows “ghost” SRAM cells
- Can be used to provide I/O Port configuration data when using RC interface

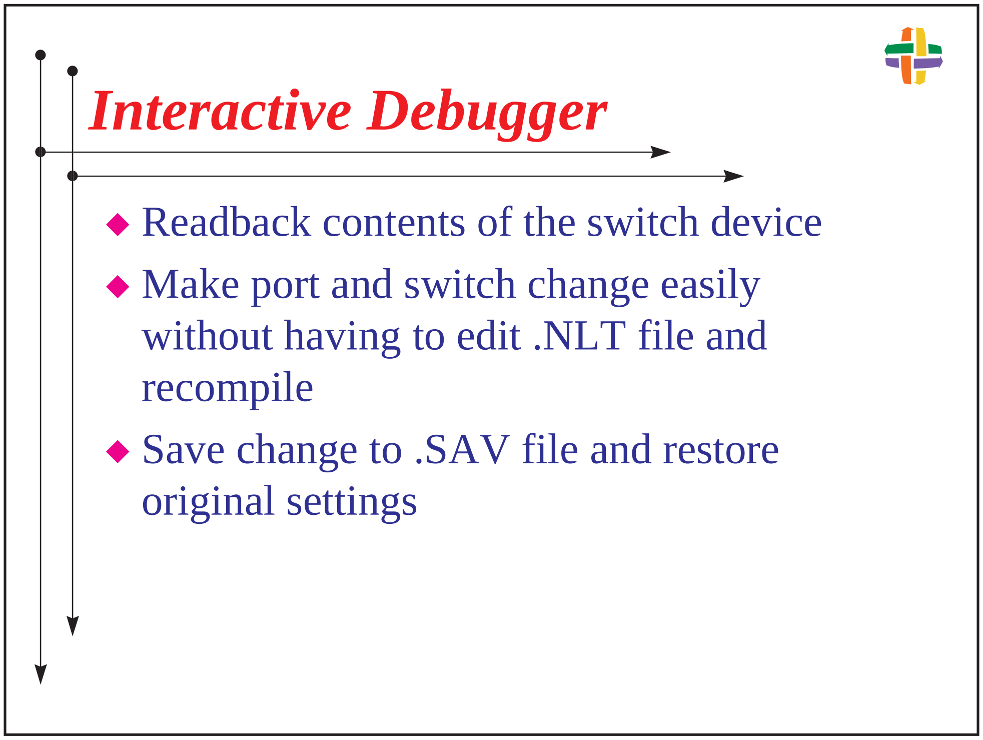
The .TBL file is a text file which shows the complete switch matrix map and the connections that have been made.



Incremental Configuration

- First bit-stream loaded initializes I/O Blocks and initial switch settings.
- IDS100/200 supports ***Incremental Compilation*** option
  - User specifies full or incremental configuration
  - Generate bitstreams for changing specified connections only
  - Other connections remain intact

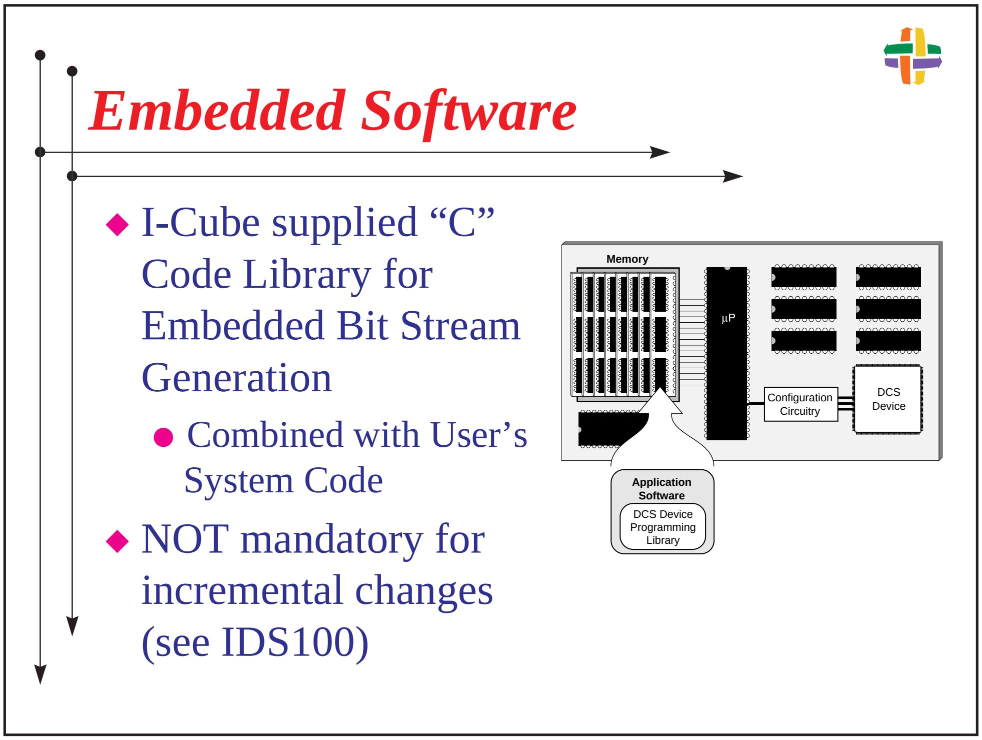
In addition to the bitstream that is used to configure the I-Cube device at power on, a number of smaller bitstreams may be created as ‘overlay’ files. These smaller files contain the data for making specific, incremental switch changes, without affecting other configuration settings.



Interactive Debugger

- Readback contents of the switch device
- Make port and switch change easily without having to edit .NLT file and recompile
- Save change to .SAV file and restore original settings

This tool is helpful during the debug phase of design. The user can read back the device contents via the JTAG port and PC printer port. This data can be modified, saved, and downloaded to the device, allowing quick and easy verification of the design.



Embedded Software

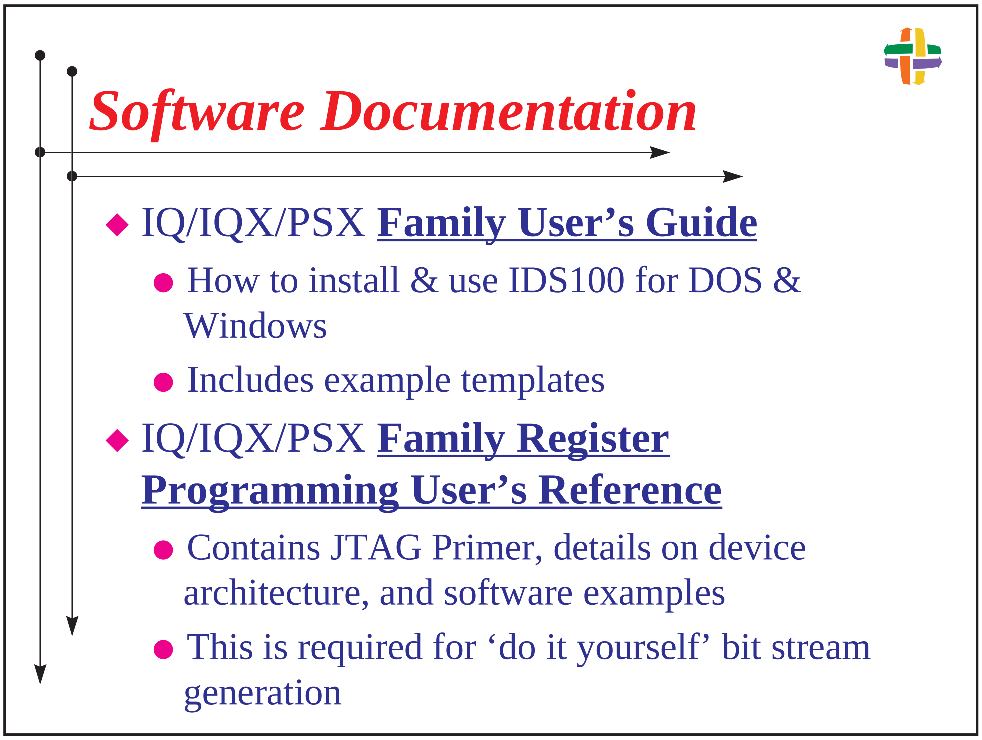
- I-Cube supplied “C” Code Library for Embedded Bit Stream Generation
  - Combined with User’s System Code
- NOT mandatory for incremental changes (see IDS100)

Memory
µP
Configuration Circuitry
DCS Device
Application Software
DCS Device Programming Library

Another software option offered by I-Cube is for the embedded applications. Using the I-Cube supplied library of function calls you can quickly develop your own code that will run on a processor or controller in your embedded system. This code will generate the bit stream fragments as needed to alter I/O port configuration or connections in the Switch Matrix.

The IDS500 library is supplied as a C source and comes as a royalty-free license. You only pay once when you purchase the library. This is available only for the IQ family.

To support the IQX family I-Cube will provide C-code example (source code) free of charge. These examples can be used with the users own code. Contact I-Cube applications for further details.



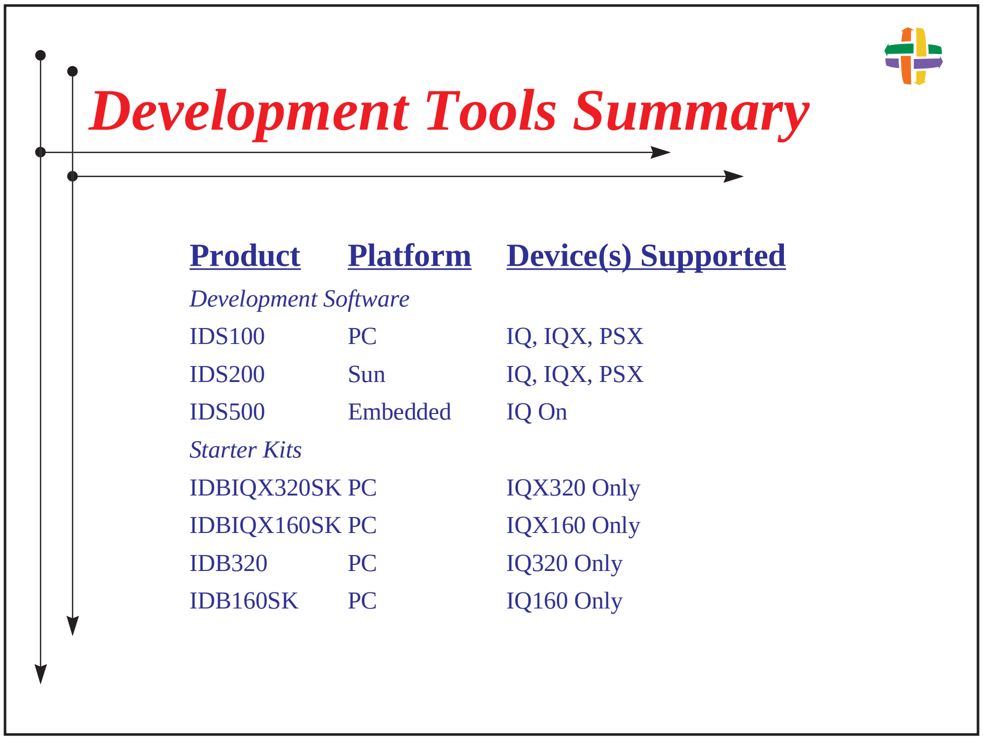
Software Documentation

- IQ/IQX/PSX **Family User’s Guide**
  - How to install & use IDS100 for DOS & Windows
  - Includes example templates
- IQ/IQX/PSX **Family Register Programming User’s Reference**
  - Contains JTAG Primer, details on device architecture, and software examples
  - This is required for ‘do it yourself’ bit stream generation

The “do it yourself” option allows you to write your own code to generate the bit stream. You require some knowledge about the PSID architecture, especially as it relates to the organization of I/O and switch configuration cells.

This approach usually results in most optimized code for your particular application. The resulting code is compact and executes very fast.

When using this option, you do not need any software from I-Cube.

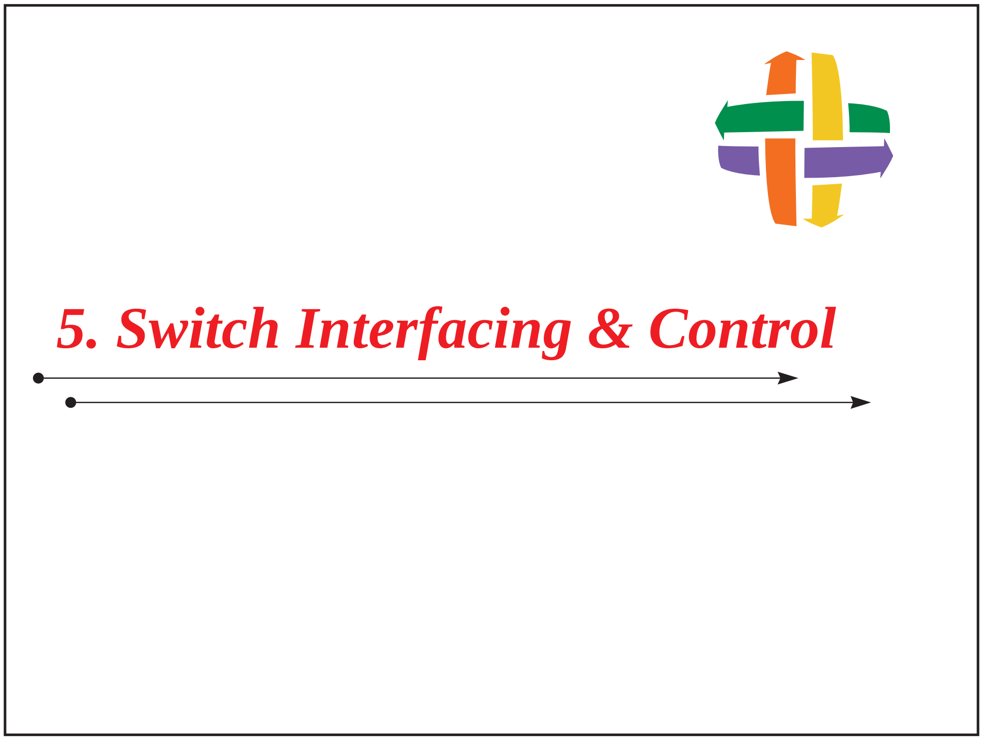


Development Tools Summary

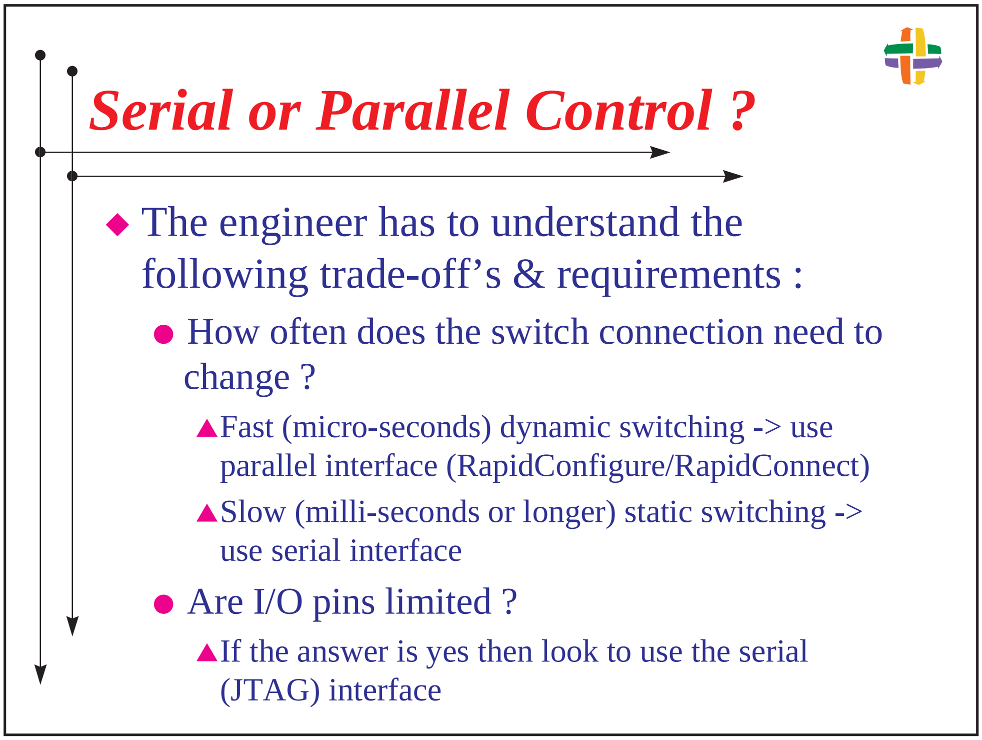
Product	Platform	Device(s) Supported
Development Software		
IDS100	PC	IQ, IQX, PSX
IDS200	Sun	IQ, IQX, PSX
IDS500	Embedded	IQ On
Starter Kits		
IDBIQX320SK	PC	IQX320 Only
IDBIQX160SK	PC	IQX160 Only
IDB320	PC	IQ320 Only
IDB160SK	PC	IQ160 Only

The slide above summarizes the development tools available from I-Cube.

Note: IDS500 only supports the IQ family. C-code examples (source code) are available free of charge to support the IQX family.



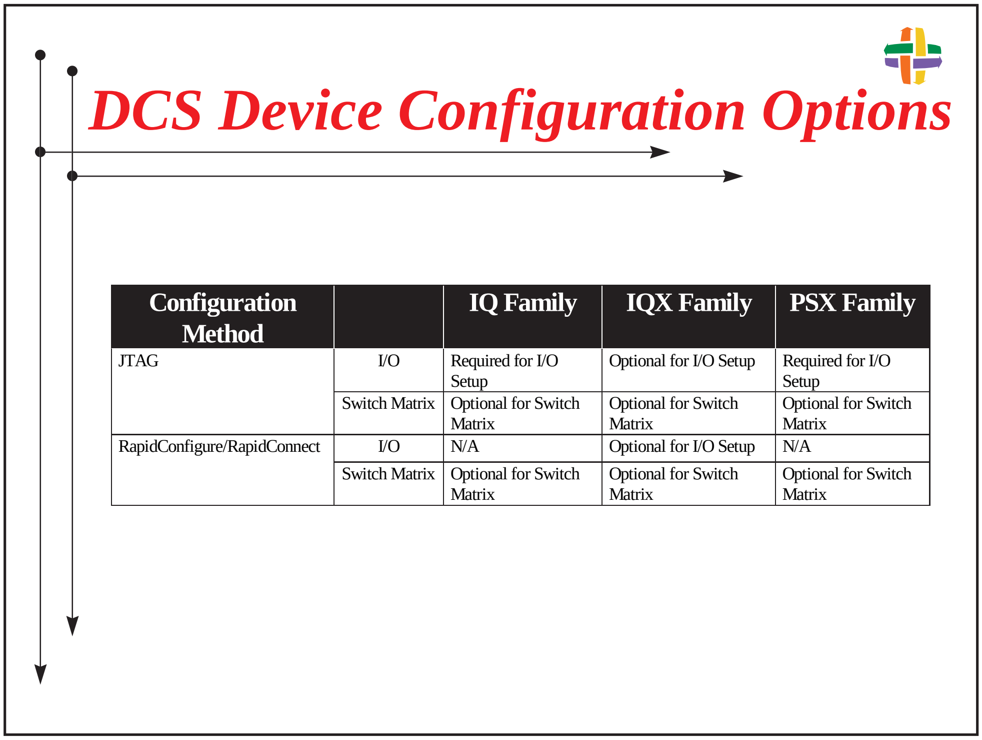
5. Switch Interfacing & Control



Serial or Parallel Control ?

- ◆ The engineer has to understand the following trade-off's & requirements :
  - ● How often does the switch connection need to change ?
    - ▲Fast (micro-seconds) dynamic switching -> use parallel interface (RapidConfigure/RapidConnect)
    - ▲Slow (milli-seconds or longer) static switching -> use serial interface
  - ● Are I/O pins limited ?
    - ▲If the answer is yes then look to use the serial (JTAG) interface

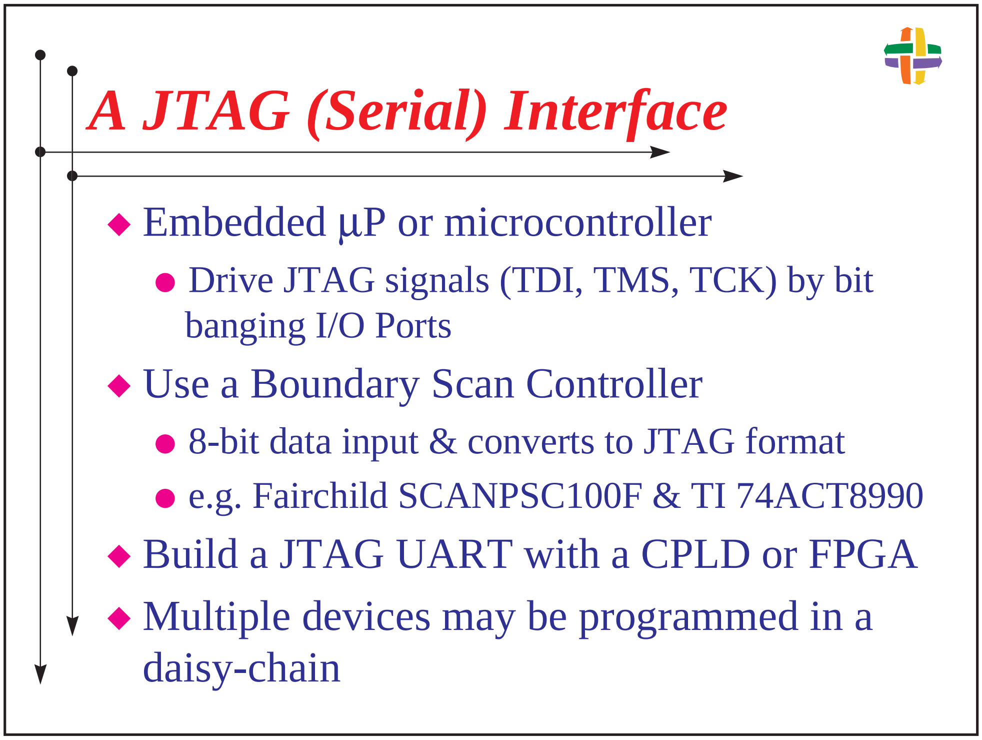
The comments above are the initial factors in deciding between serial or parallel switch interface control. The next page looks at each product family in greater detail.



DCS Device Configuration Options

Configuration Method		IQ Family	IQX Family	PSX Family
JTAG	I/O	Required for I/O Setup	Optional for I/O Setup	Required for I/O Setup
	Switch Matrix	Optional for Switch Matrix	Optional for Switch Matrix	Optional for Switch Matrix
RapidConfigure/RapidConnect	I/O	N/A	Optional for I/O Setup	N/A
	Switch Matrix	Optional for Switch Matrix	Optional for Switch Matrix	Optional for Switch Matrix

The table above summarizes the configuration requirements and options for each of the product families. This table should be used in the decision process to determine the optimal solution for a particular application.



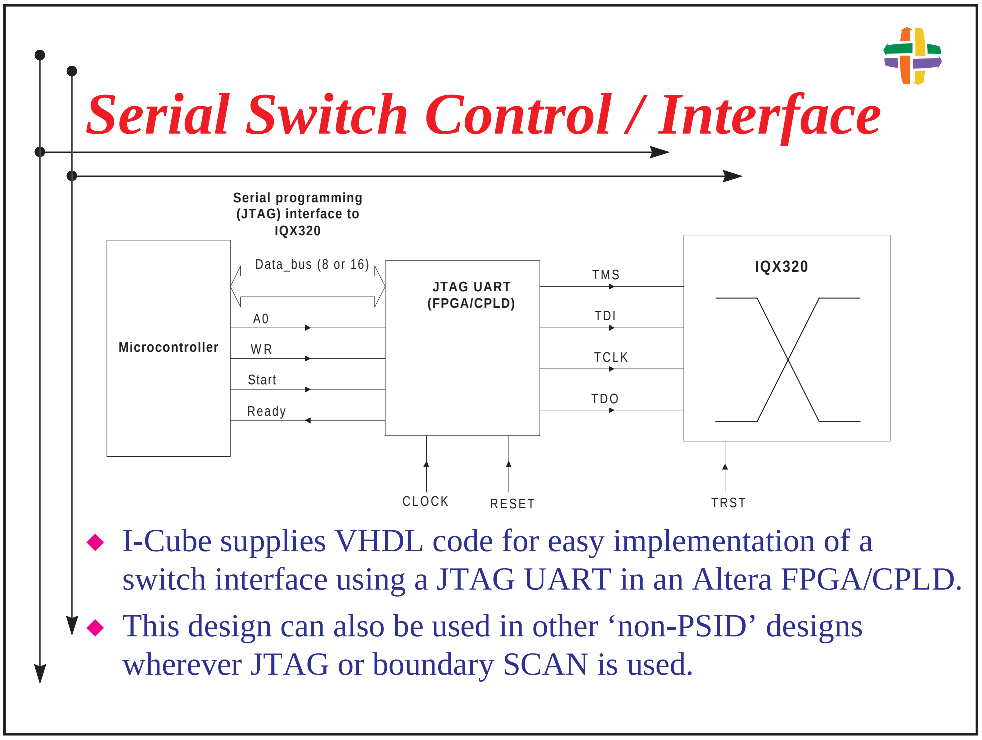
A JTAG (Serial) Interface

- Embedded µP or microcontroller
  - Drive JTAG signals (TDI, TMS, TCK) by bit banging I/O Ports
- Use a Boundary Scan Controller
  - 8-bit data input & converts to JTAG format
  - e.g. Fairchild SCANPSC100F & TI 74ACT8990
- Build a JTAG UART with a CPLD or FPGA
- Multiple devices may be programmed in a daisy-chain

An 8-bit microcontroller can be used to read a bitstream from a non-volatile memory and then to output the data to the JTAG port using the TMS, TDI and TCK pins. I-Cube devices require two bit streams, TDI and TMS, which are typically multiplexed together in memory and the read out interleaved with one another. The microcontroller must demultiplex them before outputting them separately into the TDI and TMS pins of the switch device.

•Refer to Fairchild and TI for data on Boundary Scan Controller products.

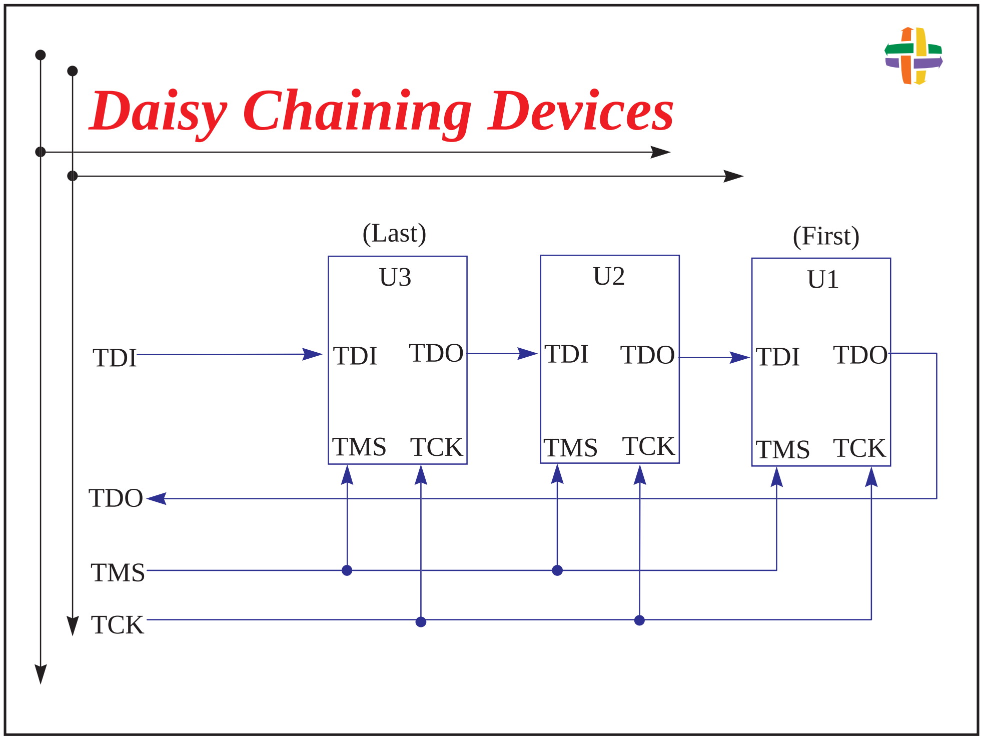
•Refer to the I-Cube application notes for details of implementing a JTAG in a CPLD or FPGA.



Serial Switch Control / Interface

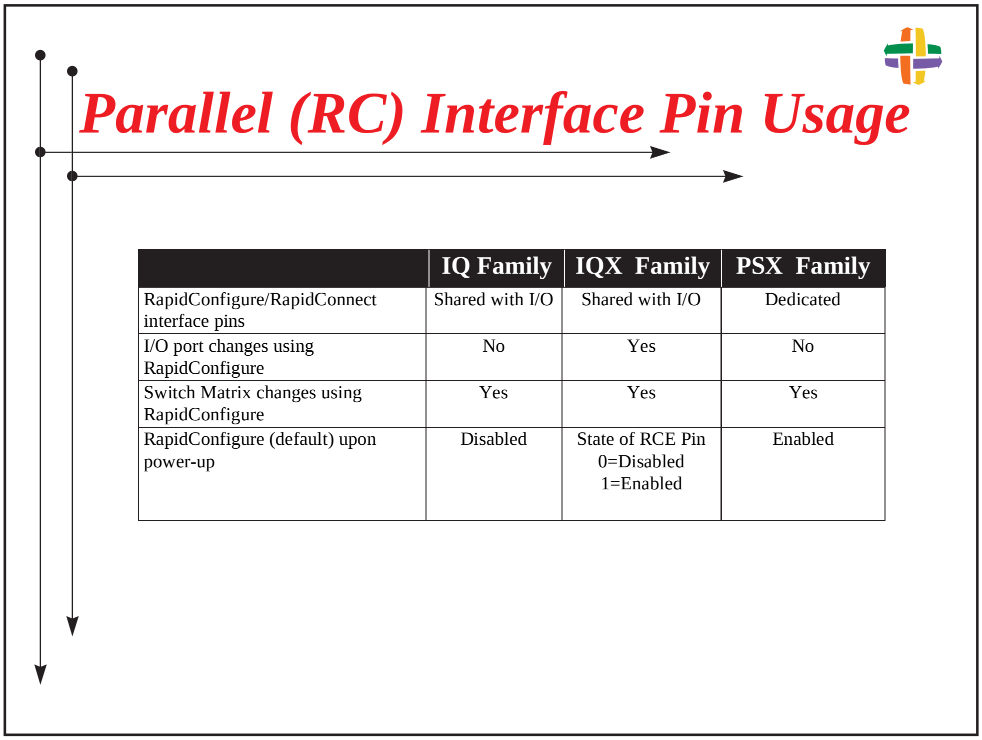
- I-Cube supplies VHDL code for easy implementation of a switch interface using a JTAG UART in an Altera FPGA/CPLD.
- This design can also be used in other ‘non-PSID’ designs wherever JTAG or boundary SCAN is used.

Refer to the I-Cube application note “JTAG UART for Controlling PSIDs” for a complete design for a JTAG UART.



Daisy Chaining Devices

Several devices may be daisy-chained together with TDO of one driving TDI of the next one.



Parallel (RC) Interface Pin Usage

	IQ Family	IQX Family	PSX Family
RapidConfigure/RapidConnect interface pins	Shared with I/O	Shared with I/O	Dedicated
I/O port changes using RapidConfigure	No	Yes	No
Switch Matrix changes using RapidConfigure	Yes	Yes	Yes
RapidConfigure (default) upon power-up	Disabled	State of RCE Pin 0=Disabled 1=Enabled	Enabled

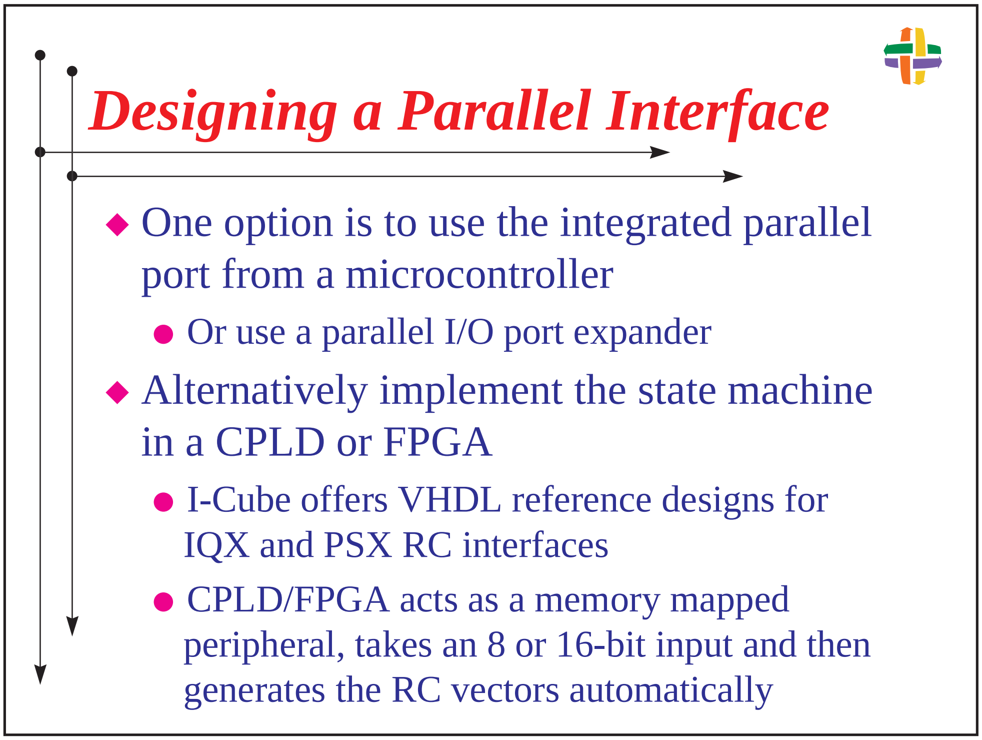
The table above summarizes pin usage among the product families.

Note how only the PSX family has dedicated RC pins. Also note that only the IQX family can change I/O port settings using RC interface.

<u>RC Notation</u>

IQ, PSX: RC=RapidConnect

IQX: RC=RapidConfigure



Designing a Parallel Interface

- One option is to use the integrated parallel port from a microcontroller
  - Or use a parallel I/O port expander
- Alternatively implement the state machine in a CPLD or FPGA
  - I-Cube offers VHDL reference designs for IQX and PSX RC interfaces
  - CPLD/FPGA acts as a memory mapped peripheral, takes an 8 or 16-bit input and then generates the RC vectors automatically

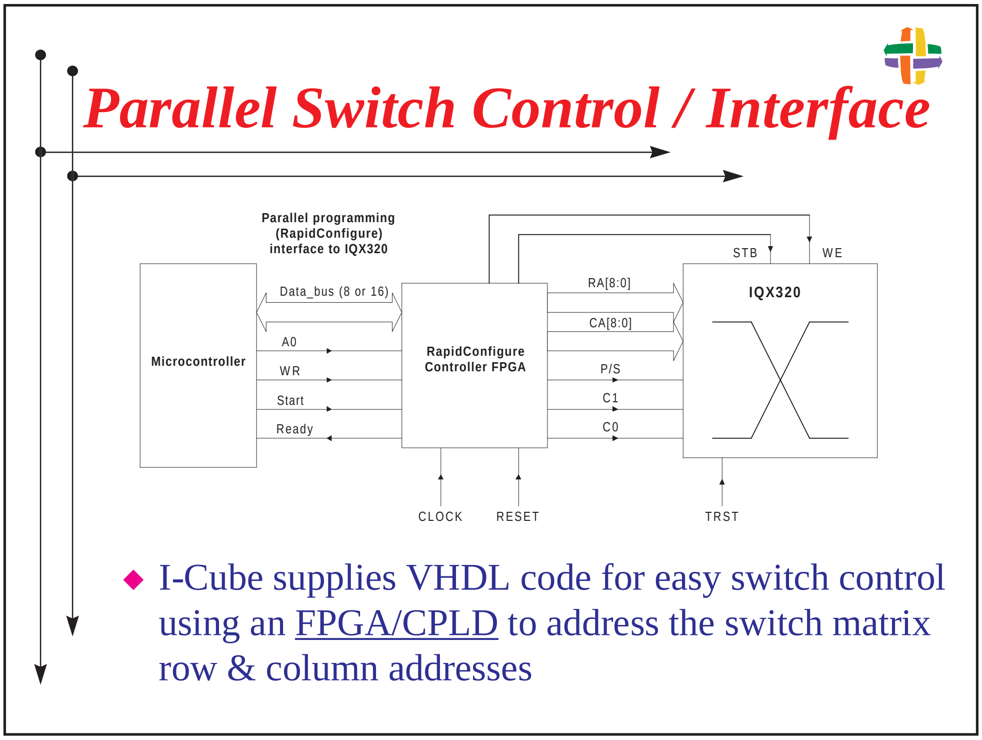
Refer to the I-Cube application notes:

"Designing a 128x128 Crossbar With Fast Reconfiguration"

"Designing a Large Crosspoint With Fast Reconfiguration"

"Implementing Bank Switching Using IQX Devices"

"RapidConnect Controller for PSX Devices"



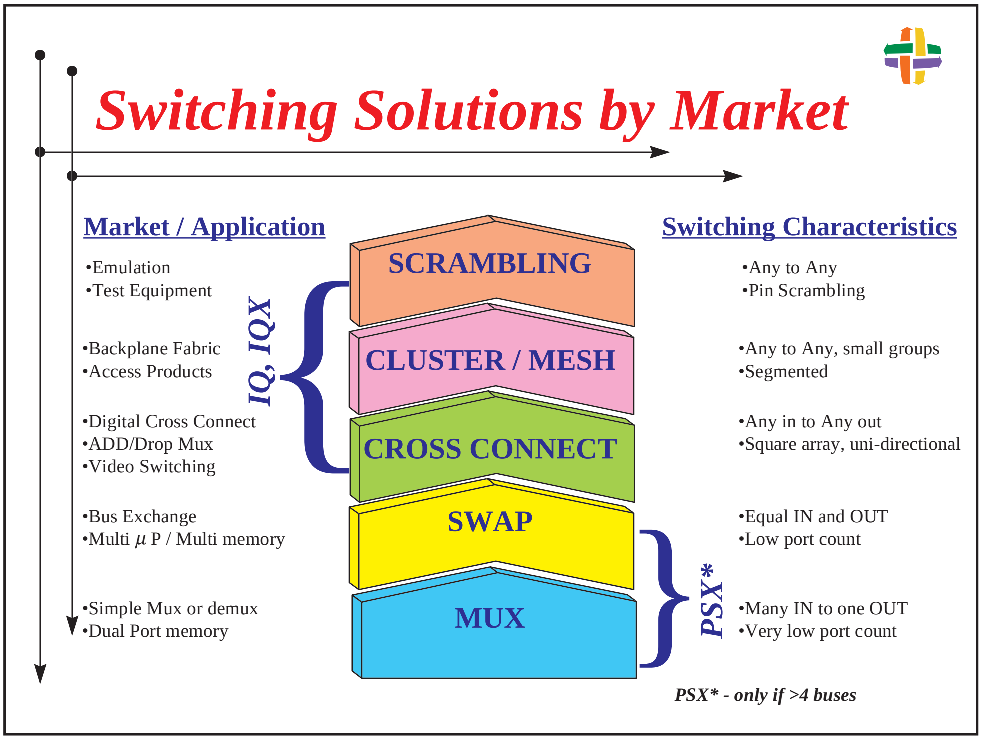
Parallel Switch Control / Interface

- I-Cube supplies VHDL code for easy switch control using an FPGA/CPLD to address the switch matrix row & column addresses

The application notes are complete “standalone” reference designs. The VHDL code implements automatic I/O port configuration, and initial switch connections. Then based on inputs from the system control bus (microcontroller) the RapidConfigure controller translates the commands to make or break a connection into the appropriate Row Address (RA) and Column Address (CA) and control (P/S, C1, C0) signals.

6. Target Markets

]

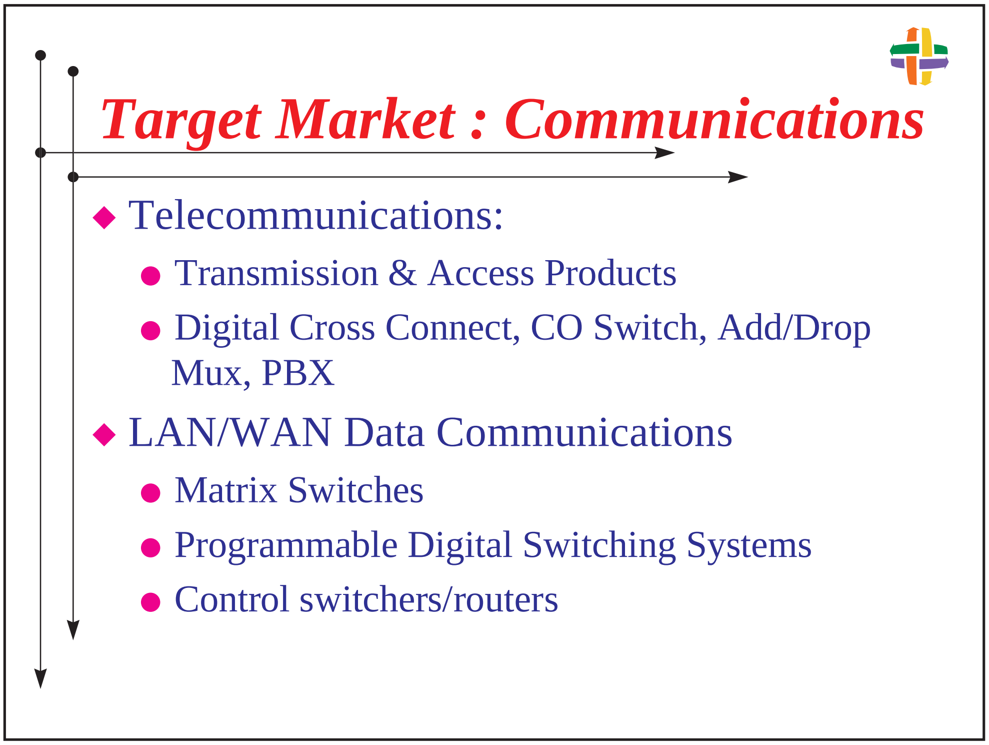


Switching Solutions by Market

Market / Application		Switching Characteristics
•Emulation •Test Equipment	SCRAMBLING	•Any to Any •Pin Scrambling
•Backplane Fabric •Access Products	CLUSTER / MESH	•Any to Any, small groups •Segmented
•Digital Cross Connect •ADD/Drop Mux •Video Switching	CROSS CONNECT	•Any in to Any out •Square array, uni-directional
•Bus Exchange •Multi μ P / Multi memory	SWAP	•Equal IN and OUT •Low port count
•Simple Mux or demux •Dual Port memory	MUX	•Many IN to one OUT •Very low port count

PSX - only if >4 buses*

DCS devices are multi-market products. Different markets and applications have varying switching characteristics as shown above. Essentially the greater the switching flexibility required, the higher the value DCS devices offer to a user.

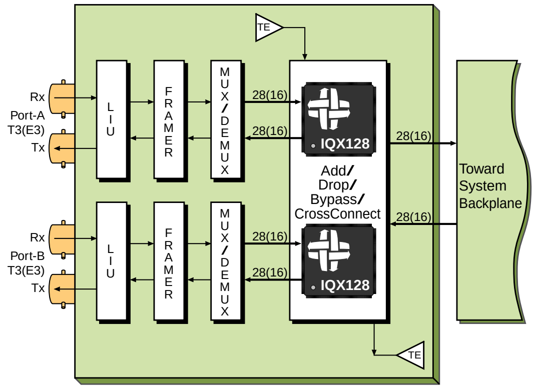


Target Market : Communications

- ◆ Telecommunications:
  - ● Transmission & Access Products
  - ● Digital Cross Connect, CO Switch, Add/Drop Mux, PBX
- ◆ LAN/WAN Data Communications
  - ● Matrix Switches
  - ● Programmable Digital Switching Systems
  - ● Control switchers/routers

I-Cube’s switching technology is a great fit in WAN, Digital Cross Connect, and Digital Access Products, performing functions such as crosspoint switching, signal routing, data aggregation and switching. Additionally DCS devices may be used to design programmable backplanes or midplanes.

Add/Drop Mux with CrossConnect

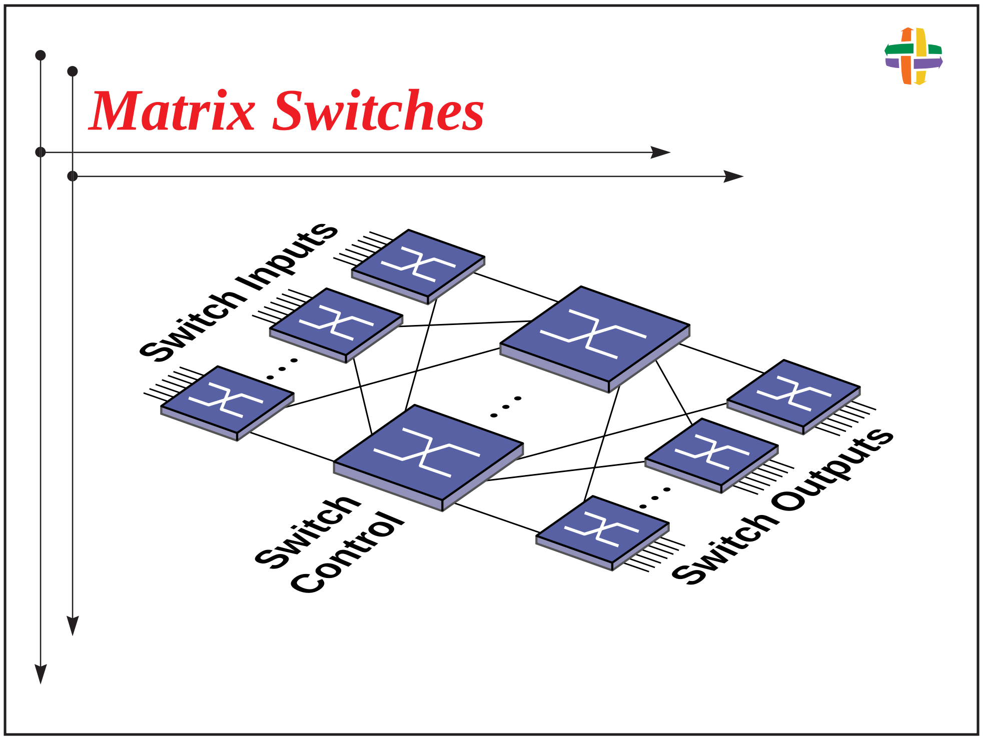


T3(E3) Add/Drop/Bypass MUX with CrossConnect

The design shows an ADD/Drop/Bypass mux with CrossConnect for high speed serial data streams such as T3 (44.736 MHz) or E3 (34.36 MHz). In such system, the digital streams of the same bit rates are received on two ports. The signals are converted into single ended TTL levels using clock recovery and frame synchronization circuits (shown as LIU and FRAMER in the figure). This data stream is then applied to a T3 or E3 mux/demux device which breaks it down to its sub-rate tributaries. As the T1 (1.544 MHz) and E1 (2.048 MHz) constitute the basic TDM data/voice circuits for T3 and E3 respectively, the mux/demux device has either 28 or 16 data and clock output and input lines.

In the add/drop/bypass mux shown in the figure, there are a total of 56 or 32 incoming lines from the two ports. Only half of them-for a total of 28 or 16 either from one or both incoming T3/E3 lines-are dropped onto the system backplane. The remaining 28 or 16 are bypassed between the two ports. This function of ADD/DROP and BYPASS for individual tributaries of the two ports and backplane can be performed efficiently with two IQX240s for T3 and two IQX128s for E3. One IQX device is used for “data” lines while the other one is used for “clock” lines. In addition to having a totally non-blocking cross-connect capability, this architecture provides the loop back for the three sets of data streams-two incoming T3 or E3 lines and backplane-for testing purposes.

In addition, a monitoring port for test signal insertion and detection for sub rate tributaries can also be accomplished to further test and debug the system. The programmable I/O ports on the IQX device allows the user to set them to a logic 1, allowing the generation of an AIS alarm signal.

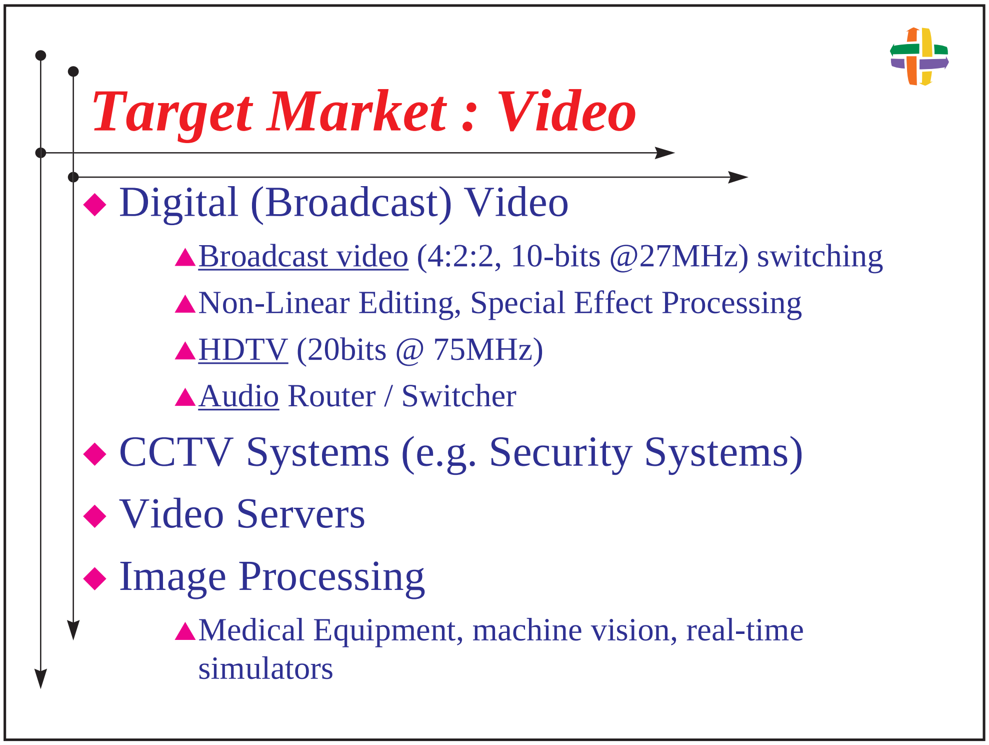


Matrix Switches

I-Cube’s IQX family are a great solution for all matrix switches, both single stage and multiple stage (e.g. CLOS networks) as found in applications such as Digital Cross Connects (DCC or DCS), LAN/WAN switches, RS232/422 data switches, etc.

The IQX products provide the ability to build a fully non-blocking solution supporting multicast or broadcast, with the benefits of higher performance, tighter skew, low pulse width distortion and lower device count.

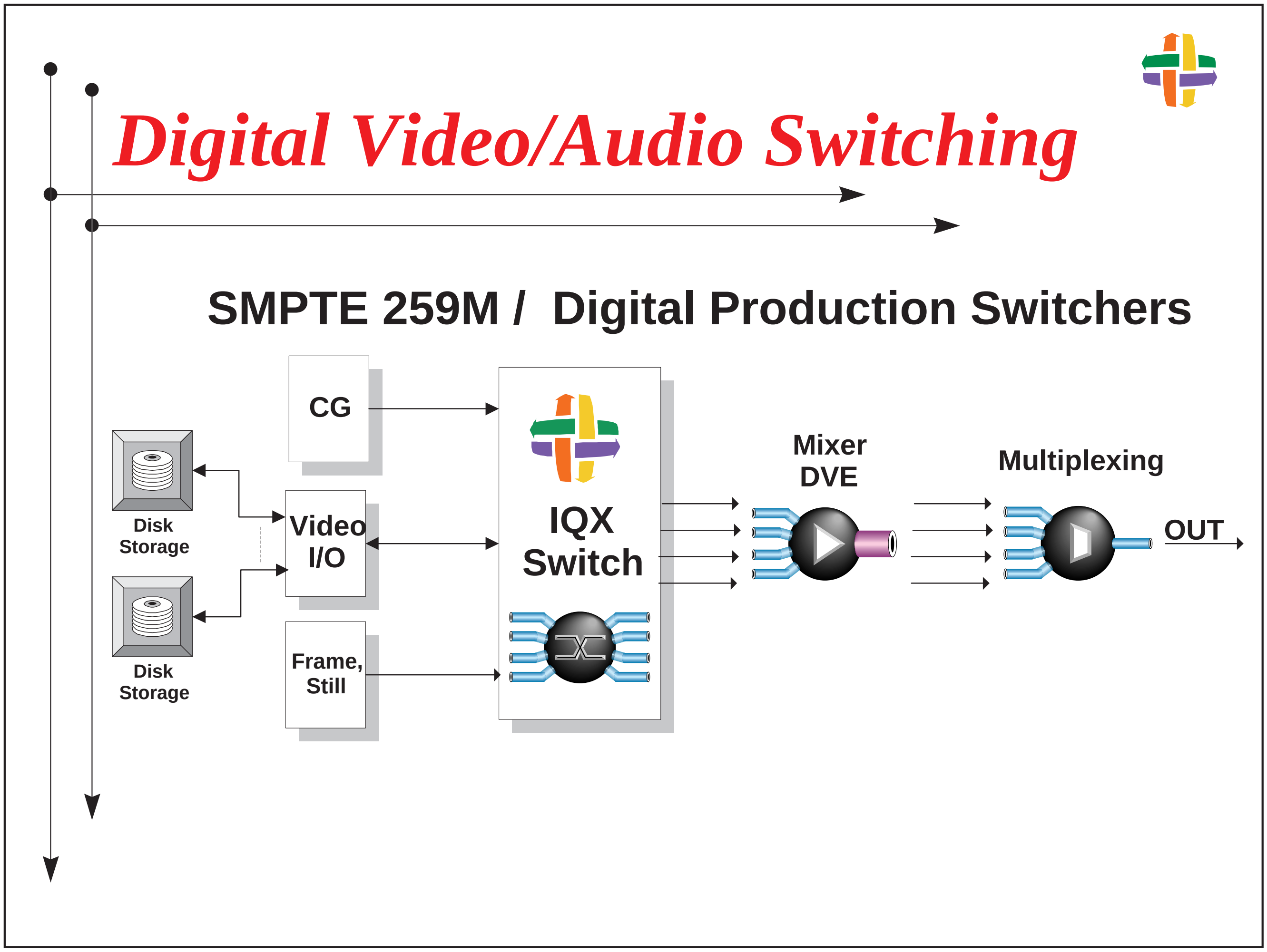
I-Cube’s SwitchSelector gives the user options on how to implement a given size of switch matrix with various IQX or IQ products, optimizing for minimum cost or minimum PCB area.



Target Market : Video

- ◆ Digital (Broadcast) Video
  - ▲Broadcast video (4:2:2, 10-bits @27MHz) switching
  - ▲Non-Linear Editing, Special Effect Processing
  - ▲HDTV (20bits @ 75MHz)
  - ▲Audio Router / Switcher
- ◆ CCTV Systems (e.g. Security Systems)
- ◆ Video Servers
- ◆ Image Processing
  - ▲Medical Equipment, machine vision, real-time simulators

I-Cube switching products are a great fit for digital video switching applications such as video servers, video editing, broadcast video switching, audio routing and image processing.



Digital Video/Audio Switching

SMPTE 259M / Digital Production Switchers

CG

Disk Storage

Video I/O

Disk Storage

Frame, Still

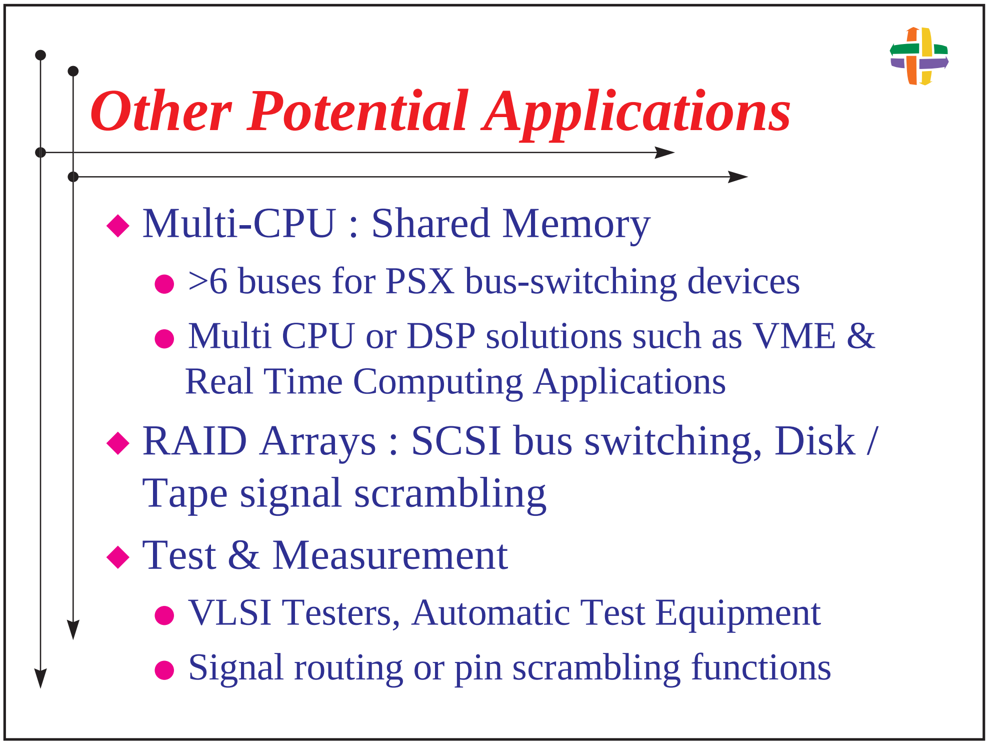
IQX Switch

Mixer DVE

Multiplexing

OUT

DCS devices provide the ideal solution for multiplexing and routing high bandwidth video buses (4:2:2), as well as digital audio switching. With a range of devices offering from 128 to 320 I/O's, IQX devices provide the designer the flexibility of using a single chip or multiple chip solution. For a matrix with a large number of 10-bit buses to switch, a cost effective solution can be implemented by "bit slicing" the bus across multiple IQX devices which simplifies the switching control and reduces system cost.

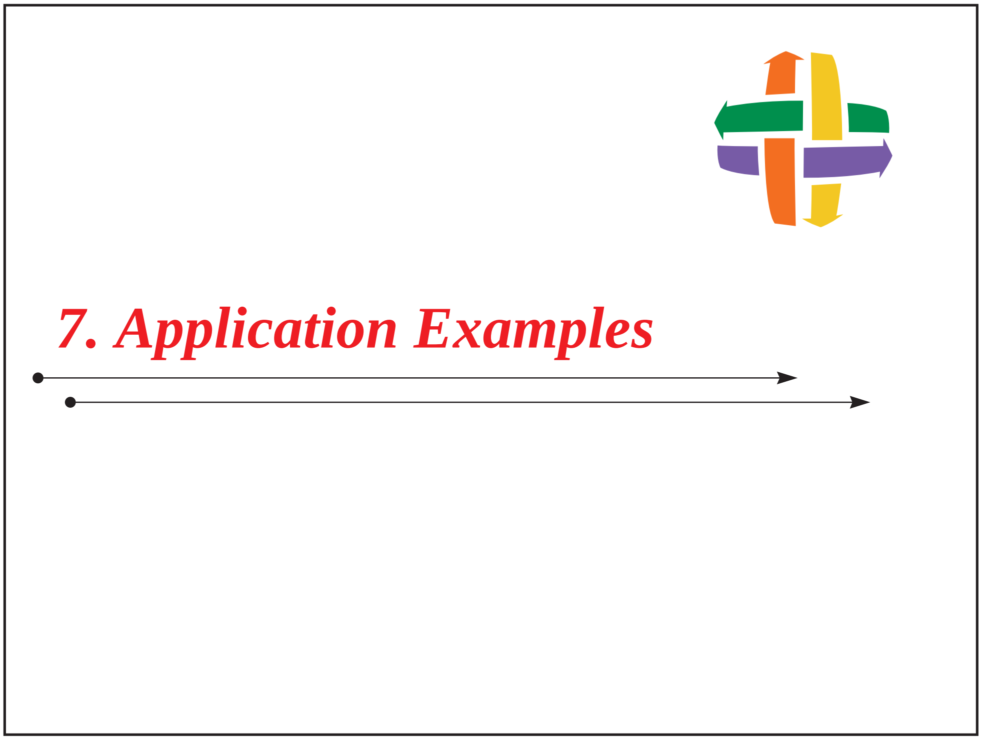


Other Potential Applications

- Multi-CPU : Shared Memory
  - >6 buses for PSX bus-switching devices
  - Multi CPU or DSP solutions such as VME & Real Time Computing Applications
- RAID Arrays : SCSI bus switching, Disk / Tape signal scrambling
- Test & Measurement
  - VLSI Testers, Automatic Test Equipment
  - Signal routing or pin scrambling functions

For high performance computing applications I-Cube's switch products offer an ideal solution for high performance bus switching and flexible backplanes.

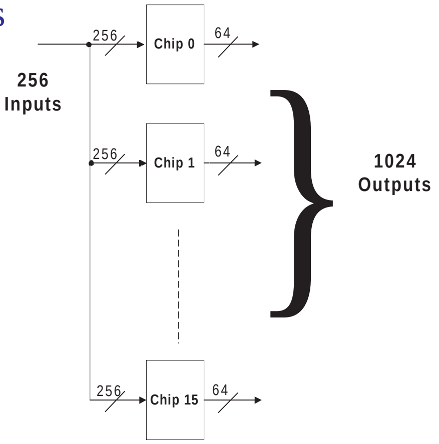
For test and measurement applications, DCS devices are an excellent solution for signal routing, pin scrambling, pattern generation, and other "programmable" functions.



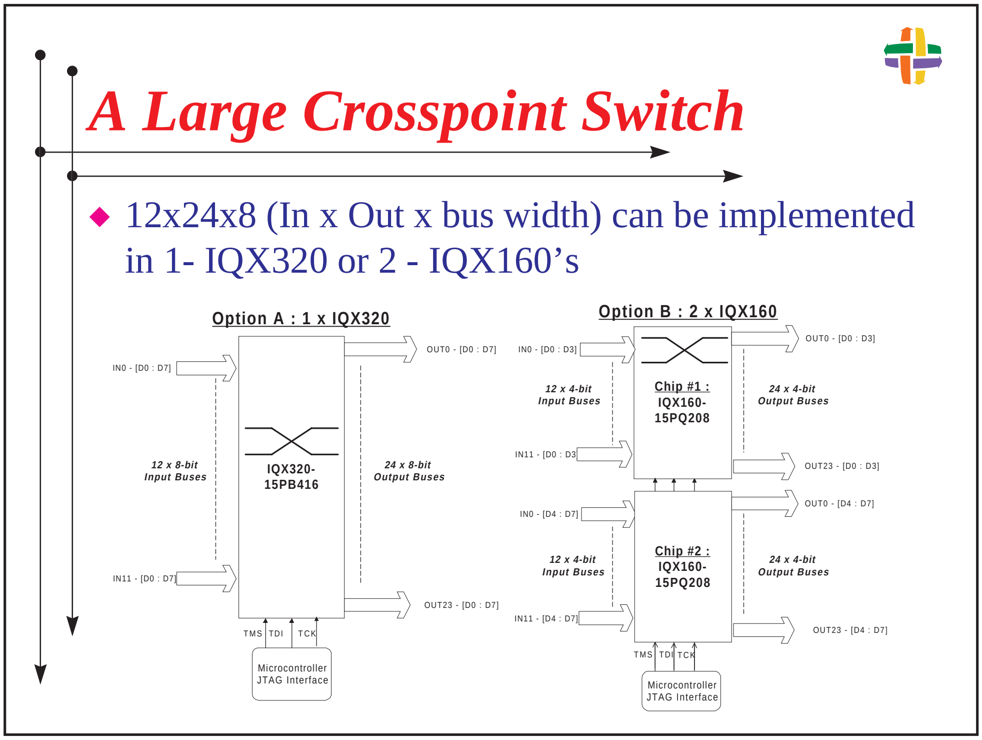
7. Application Examples

Asymmetrical Bit Switching

- I-Cube’s switch products support ‘any-in to any-out’ arrangements
- To implement a 256 x 1024 fully non-blocking switch matrix use 16 IQX320 switches, each configured as 256 x 64



This example shows how it is possible to build large asymmetrical switches easily with minimum number of devices by taking advantage of the DCS devices ability to support asymmetrical aspect ratios (different number of inputs and outputs).



A Large Crosspoint Switch

- 12x24x8 (In x Out x bus width) can be implemented in 1- IQX320 or 2 - IQX160's

12, 8-bit buses (***IN0 - IN11***) x 24, 8-bit buses (***OUT0 - OUT23***) ; treat as 36 8-bit bus switch matrix.

•IQX matrix supports *"any in x any out"* aspect ratios.

Option A : 1 x IQX320 (minimum # chips & PCB area)

Option B : 2 x IQX160's (minimum cost)

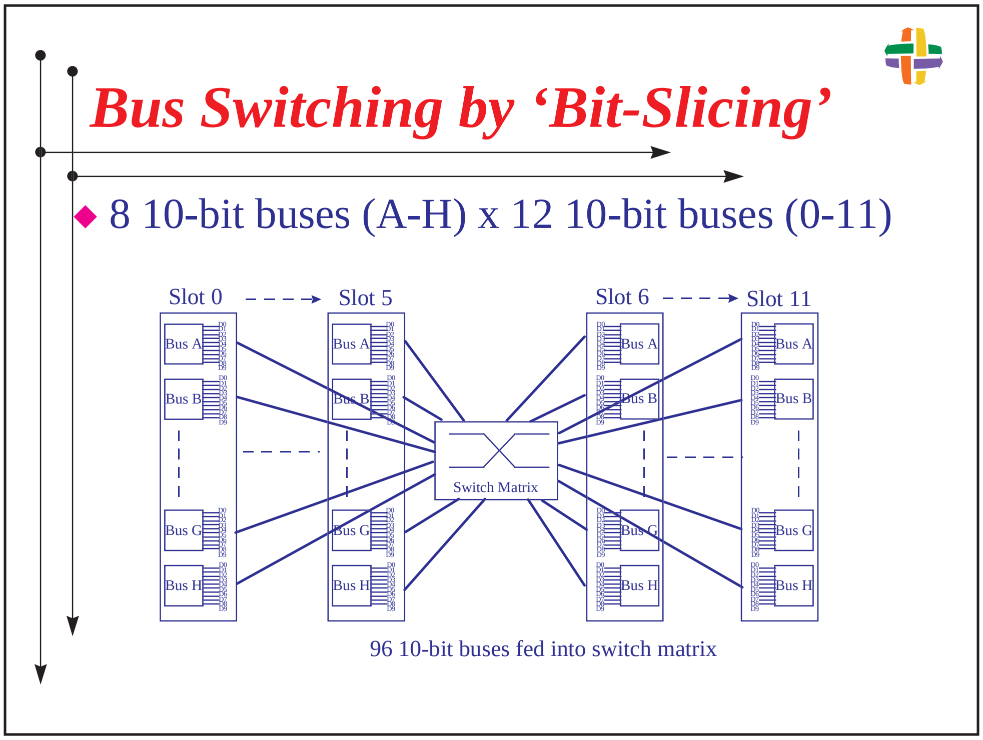
•Assume you only need to switch Bit 0's to Bit 0's, Bits 1's to Bit 1's, etc, and DO NOT connect Bit 4's to Bit 5's, etc.

•Slice outputs across multiple devices on a per bit basis and then do bus switching : Bit 0-3 on on chip #1 & bits 4-7 on chip #2

Use JTAG for configuration and switch control

•Connect multiple devices in parallel for easy switch control (since each device will have the same connections & bit stream)

Conclusion : The largest device, IQX320 provides a single chip solution for minimum PCB area. By 'bit slicing the buses' you can implement the design in multiple smaller devices (2 x IQX160) for a lower cost solution.



Bus Switching by 'Bit-Slicing'

- 8 10-bit buses (A-H) x 12 10-bit buses (0-11)

96 10-bit buses fed into switch matrix

•For implementation assume the following:

•Inputs = 48, 10-bit buses, Outputs = 48, 10-bit buses

•48x48x10 (In x Out x bus width)

•Implement switch matrix using 10 IQX128B's

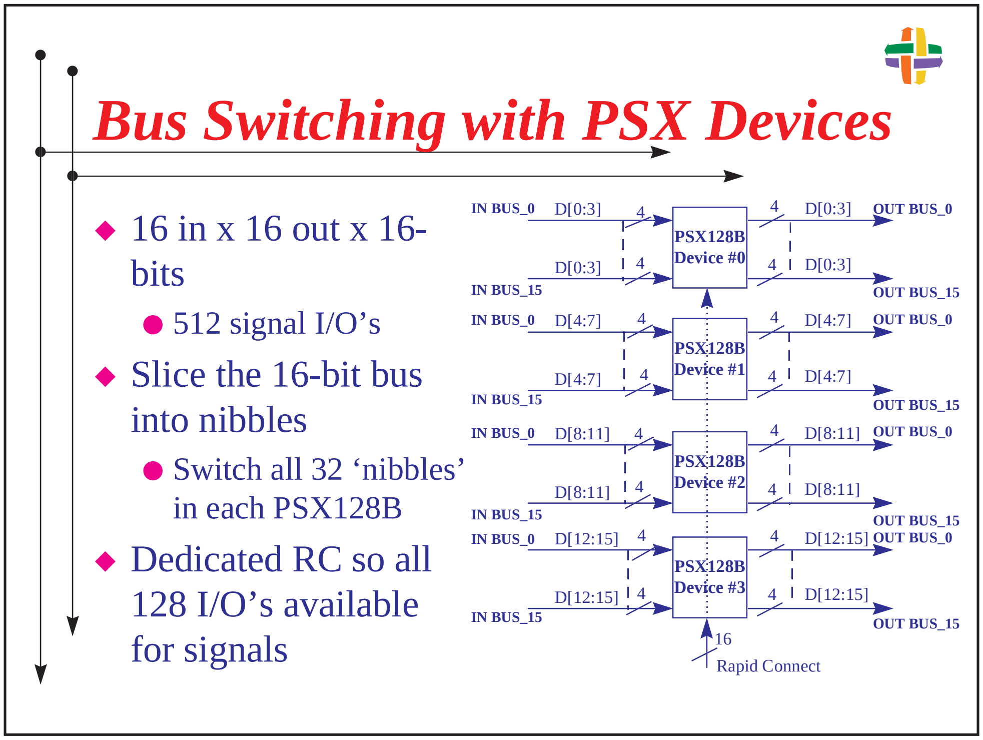
•Slice inputs and outputs across 10 devices on a per bit basis and then do bus switching

•Bits 0's to switch 0, bits 1's to switch 1, etc

•Each IQX device switch all data bit 'n''s for all 96 buses

•Reconfigure all devices simultaneously with RapidConfigure

•Switch matrix reconfigure time = 48 x 20ns = 960 nsec



Bus Switching with PSX Devices

- 16 in x 16 out x 16-bits
  - 512 signal I/O’s
- Slice the 16-bit bus into nibbles
  - Switch all 32 ‘nibbles’ in each PSX128B
- Dedicated RC so all 128 I/O’s available for signals

This example highlights the advantages of the PSX family for bus switching, especially when there are a large number of buses.

PSX advantages:

•Dedicated RapidConnect pins means all 128 I/O Ports can be used

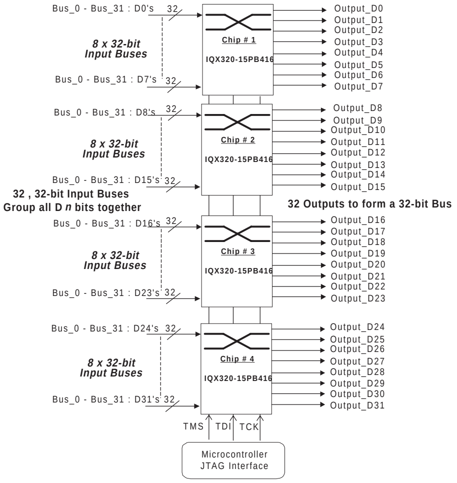
•High speed RapidConnect interface allow real-time switching (12.5ns)

•Single ‘write’ instruction can switch all four I/O’s in each nibble

•Dual configuration banks allow for synchronous update of complete switch matrix

Implementing a 32x1x32 Mux

- I-Cube’s IQX320 devices provide :
  - Many I/O’s with high performance
  - Easy to program & control
- Program at power on only
  - KEY control for mux select function



The function to be implemented is 32:1 mux for 32-bit buses, giving a total I/O = 1056.

•Use 4 IQX320’s: each device switches 8, 32-bit buses

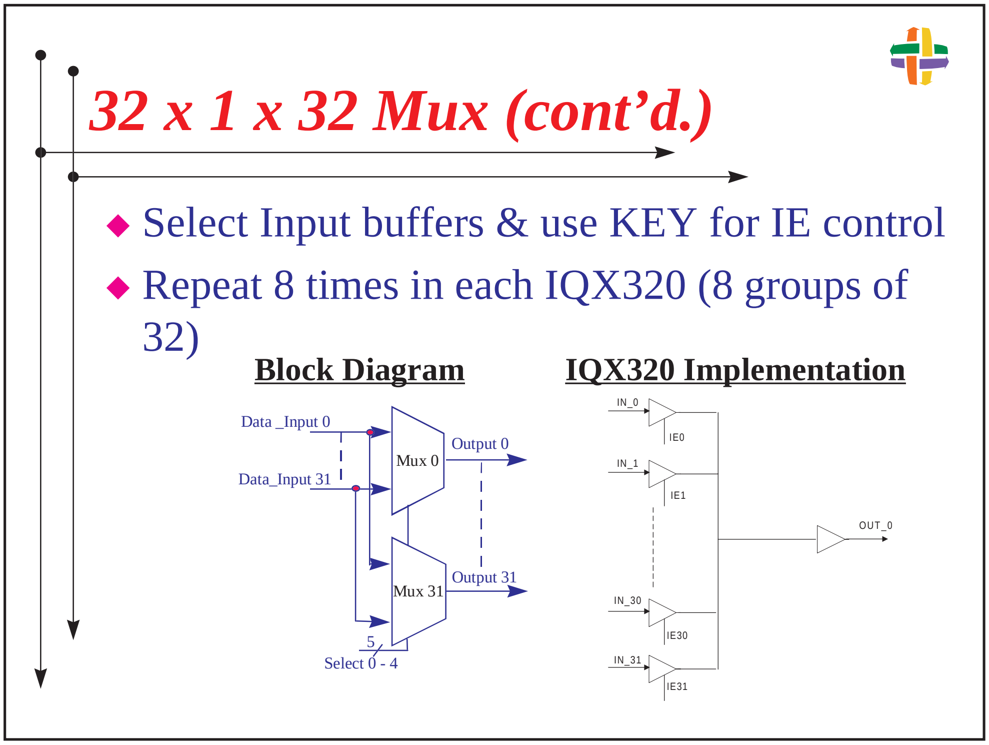
•Recognize that the D0’s only ‘switch’ to other D0’s, D1’s to D1’s etc, and that D4’s do NOT switch to D5’s

•‘Bit slice’ the buses across multiple devices

•All D0’s to D7’s go to chip #1, D8’s - D15’s to chip #2, etc.

•Since each of the 4 IQX320’s has the same configuration, you can download the same bitstream to each device.

•With just 4 devices you can implement the complete design. Programming is a single bitstream download at power up. Then the complete circuit can be controlled by just 5 input pins (KEY signals) to select the 1 of 32 inputs



32 x 1 x 32 Mux (cont'd.)

- Select Input buffers & use KEY for IE control
- Repeat 8 times in each IQX320 (8 groups of 32)

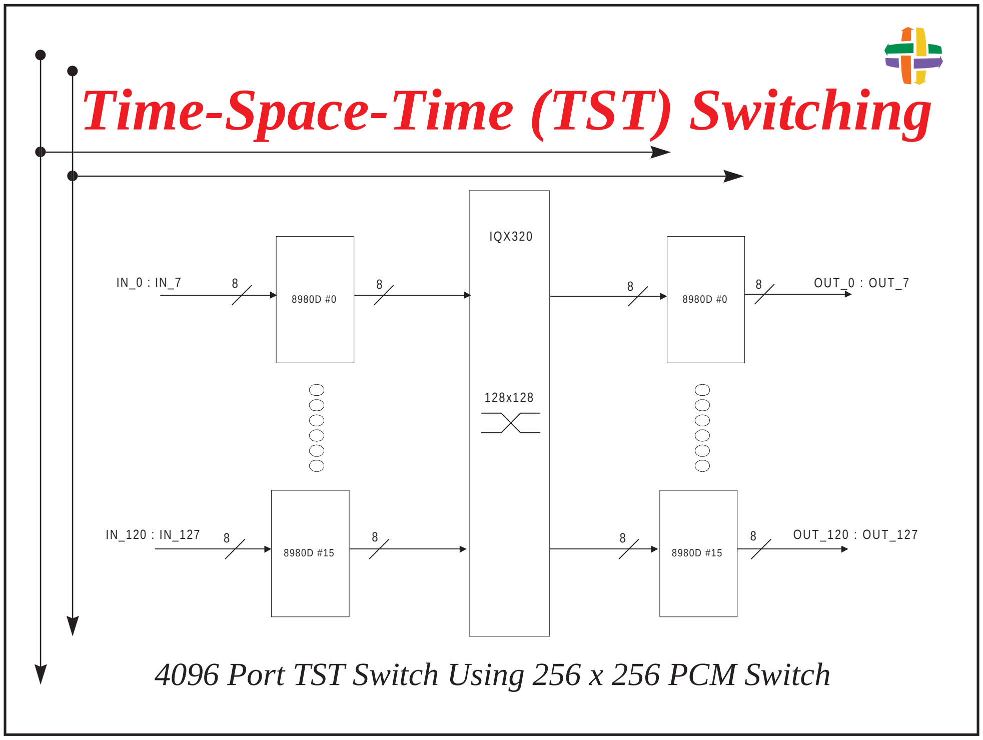
Block Diagram

IQX320 Implementation

Since this design is a mux rather than a true switch, we can take advantage of a couple of features in the IQX320 to simplify design and control of the chip: For each 32-bit group (for example all the D0's) you would specify 32 input ports to be an **IN**put buffer with a **I**nput **E**nable signal (refer to IQX datasheet for I/O port functions). These 32 input ports are connected throught the switch matrix to a common output port, and then you use the IE*n* signals to just mux a single input to the output.

This highlights the second unique feature of the IQX family; the KEY function (refer to IQX datasheet for full description). There are 5 KEY pins as inputs for the IQX320. Each I/O Port contains a 5-bit tag which can be programmed with a unique value when the I/O ports are configured. A comparator in each I/O Port continuously compares the programmed tag value with the signals present on the KEY pins. The output of the comparator can be selected as a control signal.

For this example, you define the 32 input ports to have the tag values 0..31. Then you use the 5 KEY pin to select a port between 0 - 31. Essentially the 5 Key pins are the SELECT0_4 pins shown in diagram. This has the advantage of requiring only a single programming operation at power on; after that the circuit is controlled by just 5 KEY pins.



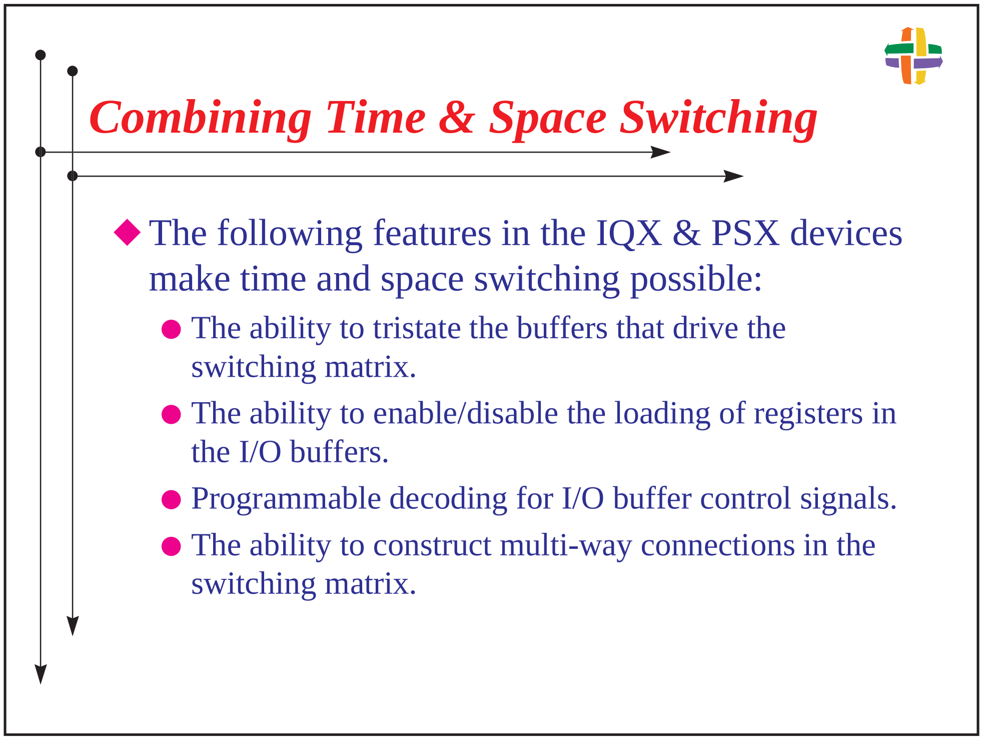
Time-Space-Time (TST) Switching

4096 Port TST Switch Using 256 x 256 PCM Switch

Time Division Multiplexing (TDM), or time switching, is used to switch continuous data streams. Space switching is used to switch entire data links. I-Cube’s Digital Crosspoint Switching (DCS) devices offer integrated, highly flexible space (crosspoint) switching solutions.

The 8980D is a 256x256 channel PCM Digital Crosspoint Switch that established a path between any input and any output over the internal ST-BUS (Serial Telecom Bus). The 8980D is available from MITEL and IMP, Inc.

This example uses the 8980D 256x256 PCM Switch and an IQX320. I-Cube’s crosspoint switches range from 32 to 320 I/O devices, providing the ability to construct different sized T-S-T switches, using various PCM switches to provide the most cost effective solution. By aggregating the 64Kb/s line you require higher performance (and cost) time switches, but this allow the use of smaller (and lower cost) space switches.



Combining Time & Space Switching

- ◆ The following features in the IQX & PSX devices make time and space switching possible:
  - ● The ability to tristate the buffers that drive the switching matrix.
  - ● The ability to enable/disable the loading of registers in the I/O buffers.
  - ● Programmable decoding for I/O buffer control signals.
  - ● The ability to construct multi-way connections in the switching matrix.

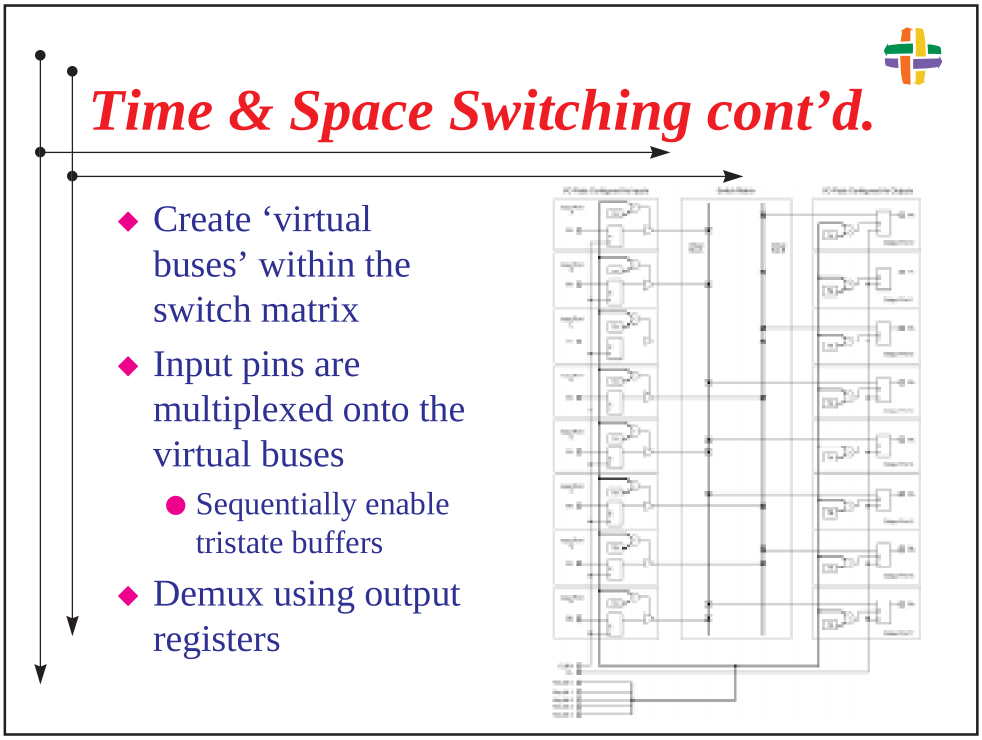
The IQX & PSX families have a unique and flexible control structure that gives the designer complete control over the behavior of each I/O block within the device. The clock, clock enable, and tristate features for each I/O buffer can be controlled by the following control resources:

•A pool of general purpose control pins.

•A 5-kit key port.

Each of the I/O buffers in the device is individually programmed to use one of these control resources to control clock, clock enable and tristate functions in that buffer.

For TDM multiplexing applications the most important feature is the key port. Each I/O port is individually programmed with a 5-bit tag value. This value is continually compared with the value present on the key port pins by a comparator located in each I/O buffer. When enabled, a match will generate a clock strobe, a clock enable and or a tristate enable.



Time & Space Switching cont'd.

- Create 'virtual buses' within the switch matrix
- Input pins are multiplexed onto the virtual buses
  - Sequentially enable tristate buffers
- Demux using output registers

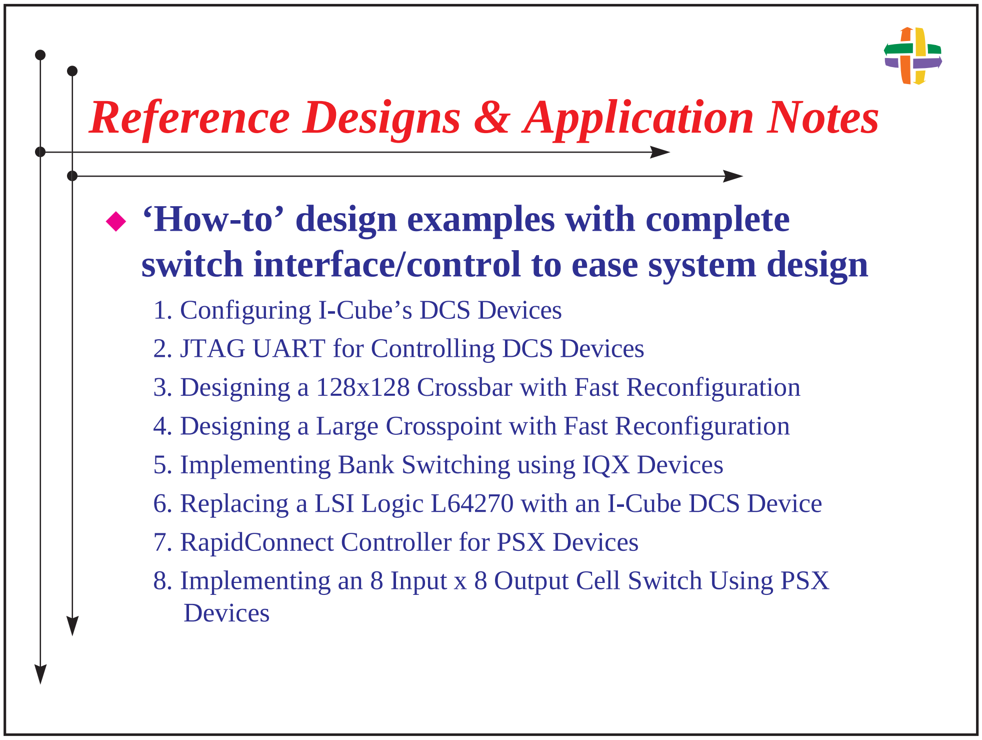
The diagram above shows how these features can be used in a combined space/time switching application. In this example, groups of four input pins and four output pins are connected together via virtual busses created within the switch matrix (refer to the Application Brief "Combining Time and Space Switching with the PSX Family" for further details).

Data from the four input pins is multiplexed onto the virtual bus by sequentially enabling the tristate buffers that drive the connections to the switch matrix. Sequencing of this operation is controlled by setting the value tag in the port to a time slot value. Each input buffer connected to the virtual bus is assigned a unique value. The time slot values are generated by a modulo 4 counter connected to the key port. This counter may be either internal or external to the device.

The TDM data stream on the virtual bus is demultiplexed to the output ports by selectively gating data into the output registers. The value programmed into the tag for each output port determines the time slot data selection. Broadcast or multicast is accomplished by programming one or more ports with the same tag value.

The assignment of an I/O port is programmable by making or breaking connections through the switch matrix.

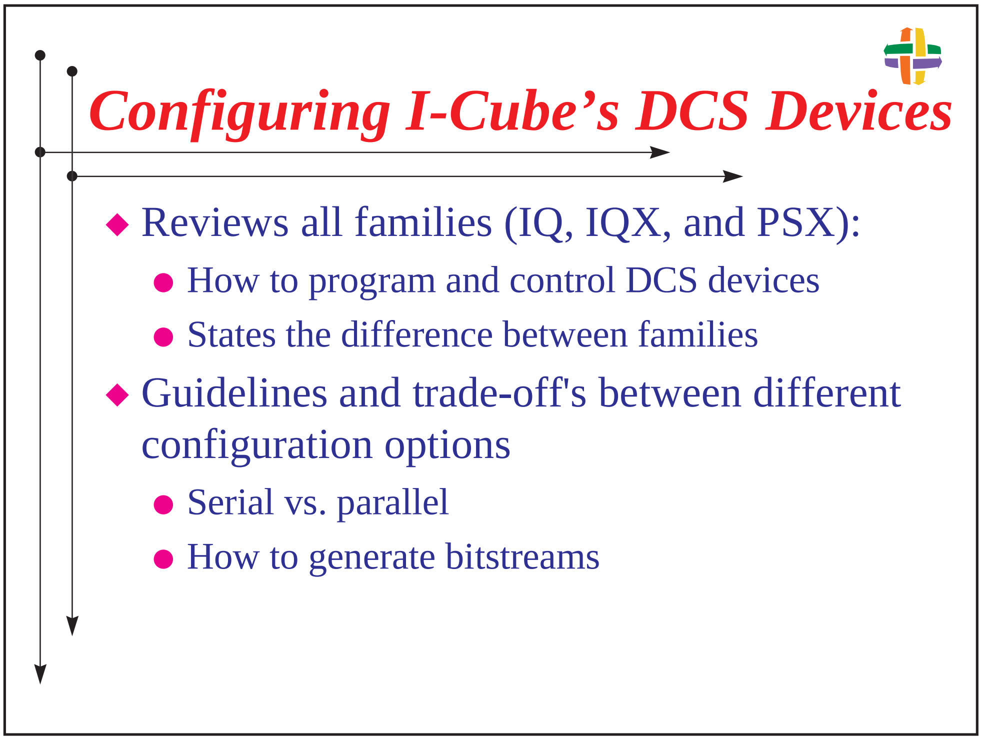
8. Reference Designs



Reference Designs & Application Notes

- ‘How-to’ design examples with complete switch interface/control to ease system design
  1. Configuring I-Cube’s DCS Devices
  2. JTAG UART for Controlling DCS Devices
  3. Designing a 128x128 Crossbar with Fast Reconfiguration
  4. Designing a Large Crosspoint with Fast Reconfiguration
  5. Implementing Bank Switching using IQX Devices
  6. Replacing a LSI Logic L64270 with an I-Cube DCS Device
  7. RapidConnect Controller for PSX Devices
  8. Implementing an 8 Input x 8 Output Cell Switch Using PSX Devices

The application notes listed above describe common designs which use I-Cube’s switch devices. Each application note is a complete system design which includes system architecture, switch matrix, as well as complete switch control interface. The switch control interface is implemented in either VHDL with example code ported to a FPGA or CPLD device.

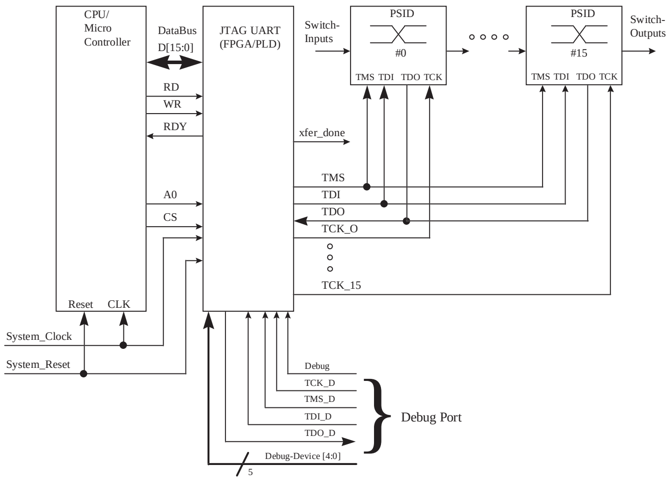


Configuring I-Cube's DCS Devices

- Reviews all families (IQ, IQX, and PSX):
  - How to program and control DCS devices
  - States the difference between families
- Guidelines and trade-off's between different configuration options
  - Serial vs. parallel
  - How to generate bitstreams

This is an excellent "DCS device primer" and discusses the basic options a user has when configuring DCS devices. After reading this application note and deciding what option to use, the user can refer to other I-Cube application notes which present practical implementation examples of both serial and parallel switch interfaces.

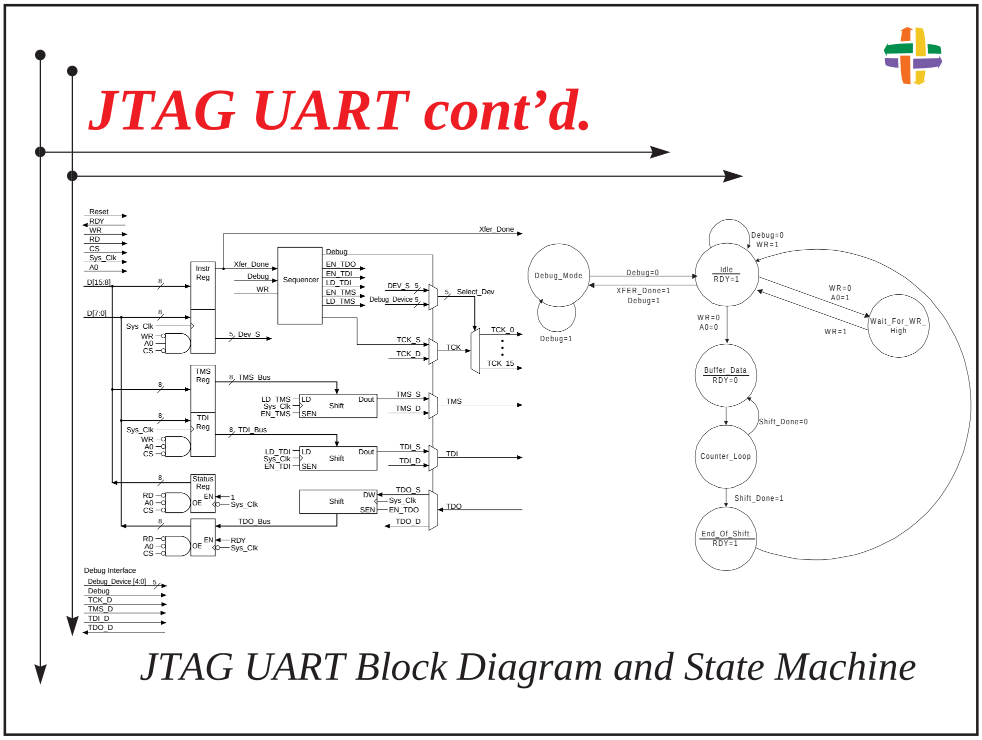
JTAG UART for Controlling DCS Devices



I-Cube's Digital Crosspoint Switching (DCS) devices have a serial interface to allow configuration of the I/O ports and switch matrix. This serial interface is compliant to the IEEE1149.1 standard, commonly referred to as JTAG. (The advantage in using the JTAG interface for controlling a DCS device, as opposed to using the parallel or RapidConfigure interface, is that it allows all the I/O Ports on the IQ and IQX devices to be used for switch inputs or outputs.)

This application note presents a design example showing how to build a "JTAG UART" using an FPGA/CPLD. This allows an easy interface between a microcontroller with an 8 or 16-bit data bus and the JTAG interface of a PSID. This example allows up to 16 separate JTAG interfaces to be supported. Additionally, the JTAG UART includes support for a debug mode, which allows the user to connect the target DCS device to the parallel port of a PC, and to control it using I-Cube's IDS100 development software running on the PC.

Note : the use of the JTAG port for power on configuration is mandatory with the IQ and PSX families, but is optional with the IQX family.



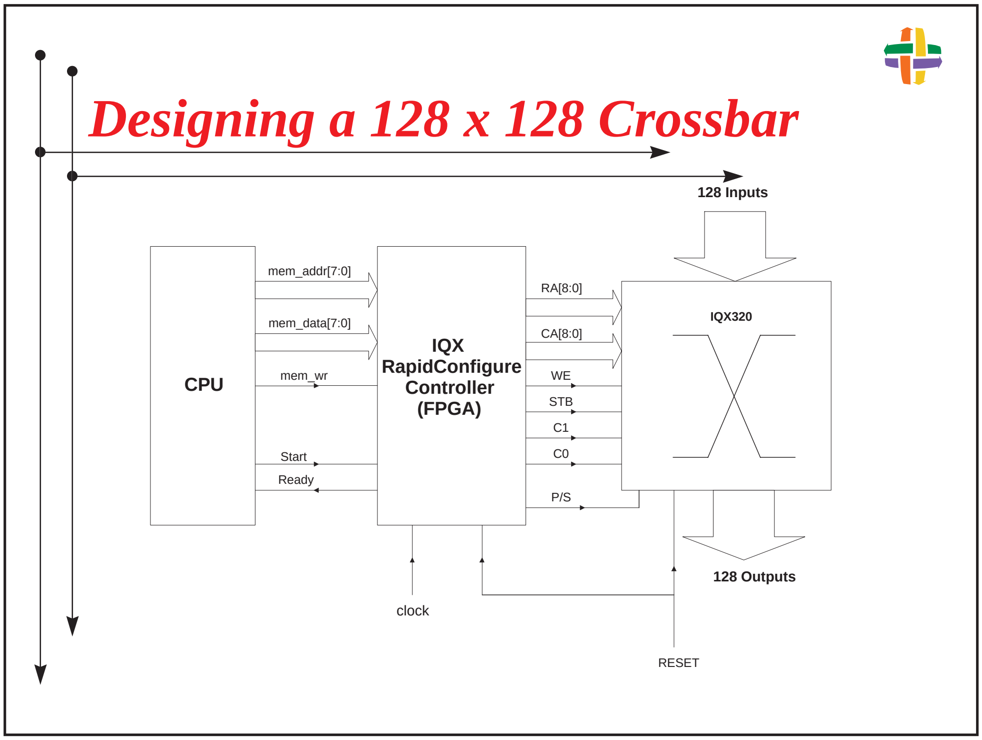
JTAG UART cont’d.

JTAG UART Block Diagram and State Machine

The diagram on the left shows the conceptual block diagrams for the JTAG UART, while the diagram on the right is the state machine which controls the operation of the JTAG UART.

JTAG UART Operation

The JTAG UART supports a 16-bit interface with a microprocessor / microcontroller. The data bus is bidirectional since both read and write operations are supported. The JTAG UART operation consists of three instructions: the first instruction that must be used is "Device Select"; this operation is required in order to select the target DCS device which will receive the JTAG bit streams for TDI and TMS. The second instruction is "Data Load" which is the mechanism by which the parallel data from the microcontroller is formatted into two bit streams (TDI and TMS) and then output to the DCS device target device that is to be programmed. The third instruction is "Status Read" which allows the microcontroller to read the status of the JTAG UART as well as reading data from the DCS device target device. The UART generates the JTAG clocks TCK_0 to TCK_15. This clock frequency is half of the System Clock.

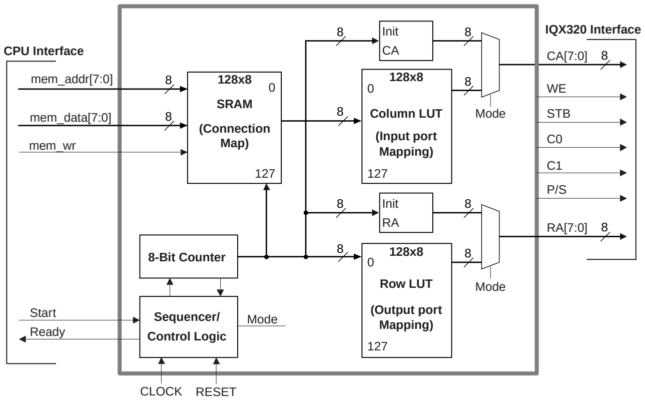


Designing a 128 x 128 Crossbar

This application note describes how to implement a 128x128 crossbar using an IQX320 device. A complete design is detailed including how to implement the RapidConfigure interface which is used for fast reconfiguration. A FPGA/CPLD based controller can be used to implement the RapidConfigure (RC) interface on the IQX320 device, and the VHDL code to implement such a controller is provided.

The reconfiguration for the complete crossbar switch is to be accomplished in under 10 µs. This is a synchronous design allowing easy implementation in FPGA/CPLD.

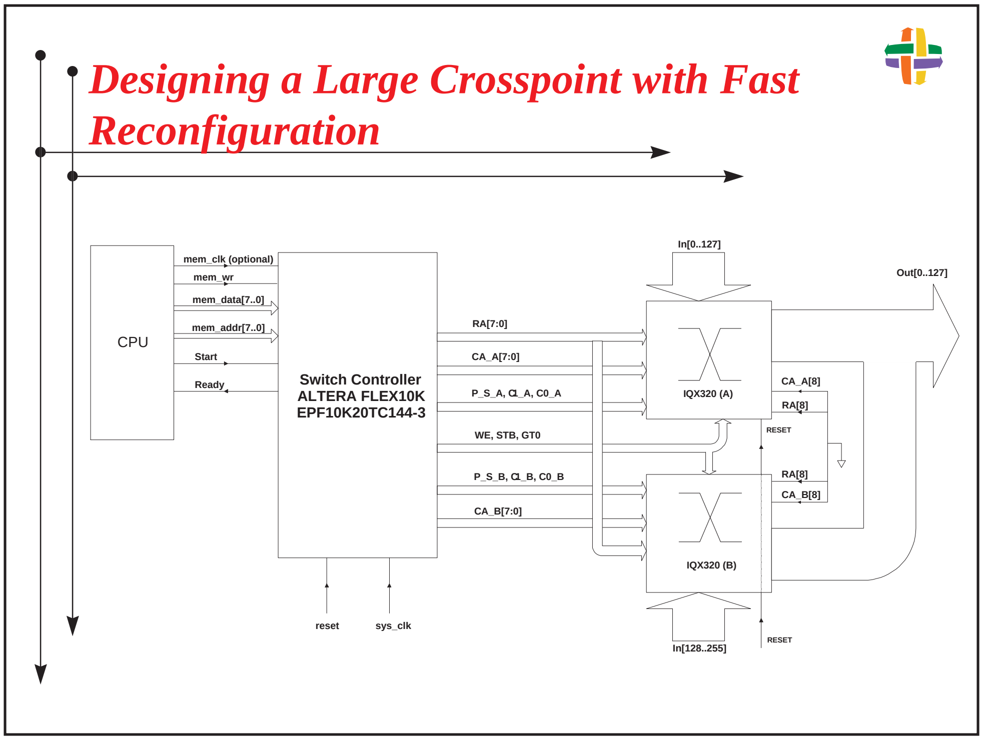
128 x 128 Crossbar cont’d.



RapidConnect Controller Block Diagram

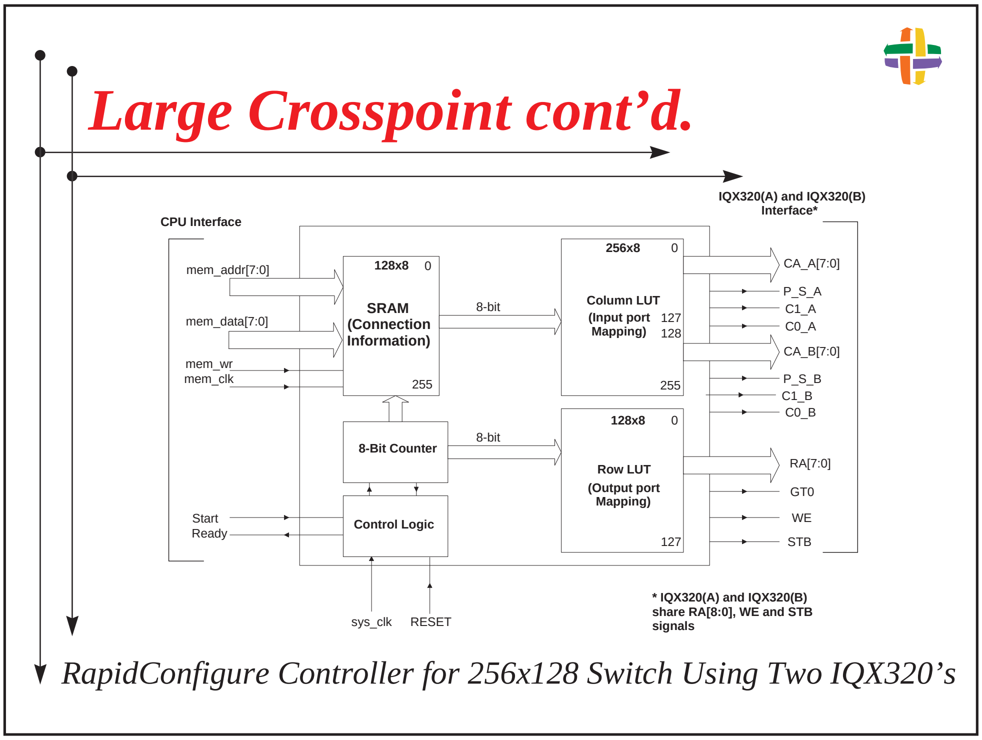
The block diagram for the switch controller is shown above. The switch controller consists of a single or dual port SRAM memory, 128 x 8 that stores the connection information, organized by the outputs in the switch. This requires 128 addresses, address 0 through 127 of the memory. For example, memory address "000h" stores the information - which is the Input Number - that the switch OUT_0 needs to connect to. Memory address "001h" stores the connection information for OUT_1 and so on. In addition, 7 locations can be used to store the I/O Configuration Holding Register values required during initialization (not used in the VHDL model).

In addition to the SRAM, the switch controller also has two 128 x 8 ROM LUTs (Look Up Tables) which gets initialized at power-on. These ROM tables contain port mapping information to map the user I/O ports to the actual IQX320 ports. ROM_RA maps the switch output ports (i.e. OUT_0 to OUT_127) to the corresponding I/O ports used as outputs for the IQX320. The ROM_RA data eventually becomes the Row Addresses RA[7:0] for the RapidConfigure interface. ROM_CA maps the switch input ports (IN_0 to IN_127) to the corresponding I/O ports used as inputs for the IQX320. The ROM_CA data eventually becomes the Column Addresses CA[7:0] for the RapidConfigure interface. User port is used as ROM address and the data corresponds to the actual IQX320 port.



Designing a Large Crosspoint with Fast Reconfiguration

The following example illustrates how a large, 256 in x 256 out crosspoint switch can be implemented using 4 IQX320 devices. This design example assumes unidirectional data flow. The reconfiguration for the entire crosspoint switch is to be accomplished in under 25µs. This switch is designed for synchronous operation.



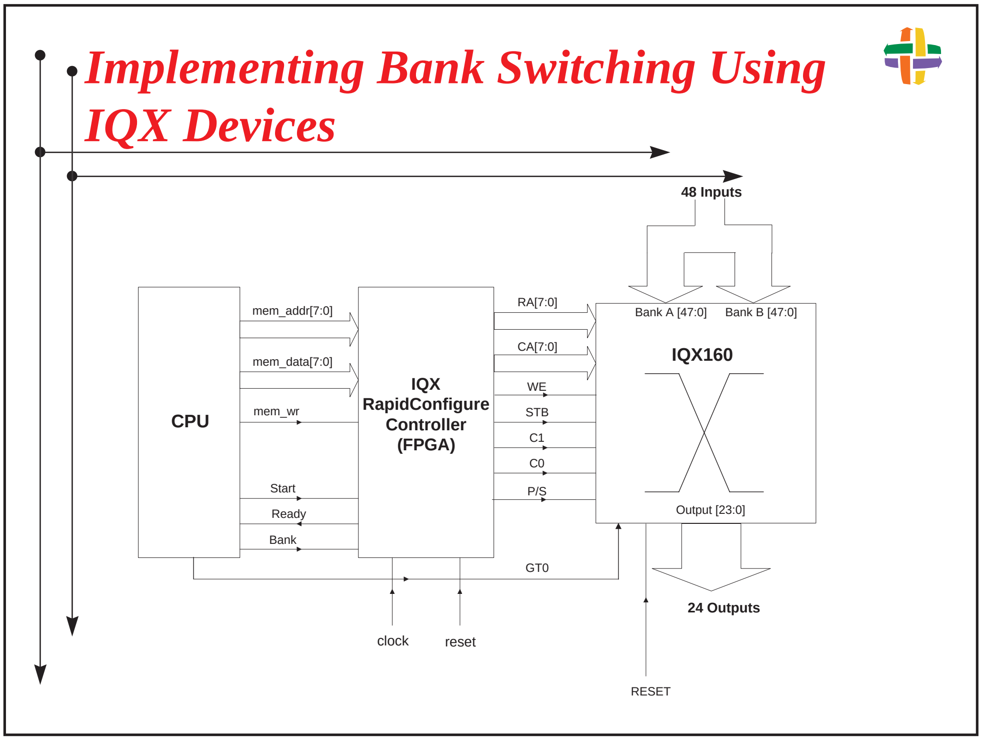
Large Crosspoint cont'd.

RapidConfigure Controller for 256x128 Switch Using Two IQX320's

The Switch controller is designed for a 256x128 switch implemented using two IQX320s. Each IQX320 device acts as a 128x128 switch and I/O ports used as outputs on both devices are tied together i.e. output_0 of device A is tied with output_0 of device B, output_1 on device A is tied with output_1 on device B and so on. This can easily extended to control a 256x256 switch implemented using 4 IQX320s.

The switch controller consists of a single or dual port SRAM memory, 256 x 8 that stores the connection information, organized by the outputs in the switch.

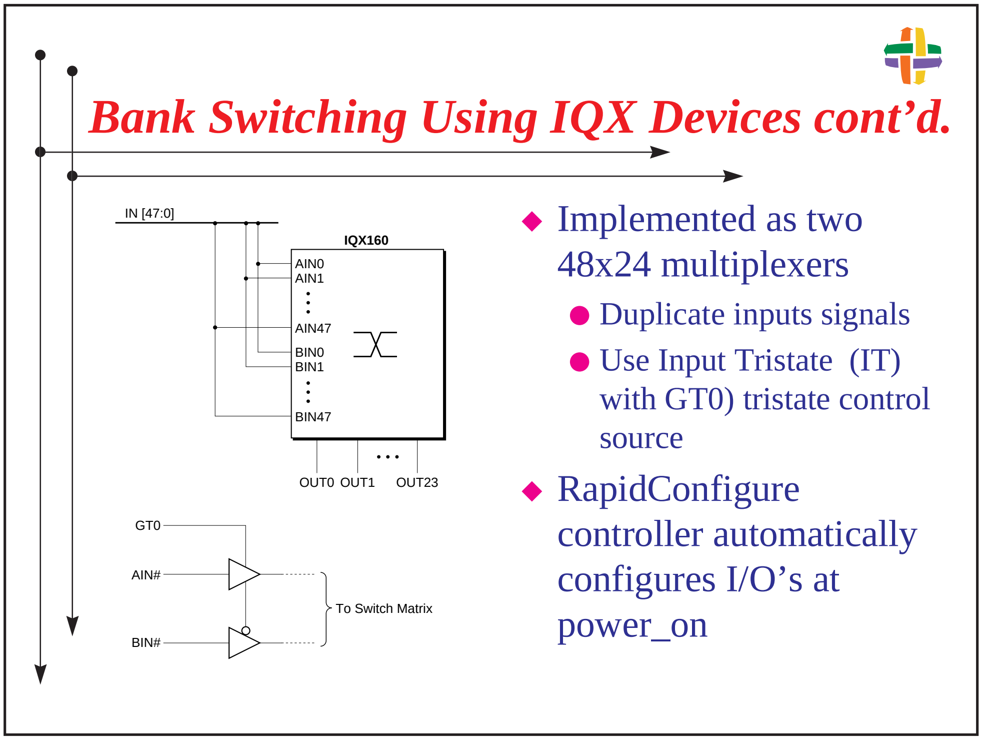
In addition to the RAM, the switch controller also has two 256 x 8 ROM (Read Only Memory) tables which gets initialized at power-on. These ROM tables contain port mapping information to map the user I/O ports to the actual IQX320 ports. ROM_Row Address maps output ports to the corresponding I/O ports used as outputs IQX320. The ROM_Row Address data eventually becomes the Row Addresses for the RapidConfigure interface. ROM_Column Address maps the input ports to the corresponding I/O ports used as input ports in IQX320. These ROM_Column Address data eventually becomes the Column Addresses for the RapidConfigure interface.



Implementing Bank Switching Using IQX Devices

Certain multiplexing and crossbar switching applications require that the entire multiplexer or switch configuration be changed instantaneously. In the PSX Bus Switching devices, the dual configuration banks allow you to pre-load the next configuration and then make it active in as little as 12.5 ns. The IQX Bit Switching devices do not offer the dual configuration banks, however, this can be accomplished using a different mechanism described in this application note.

The example implements a 48 input x 24 output multiplexer using an IQX160. A multiplexer input can be selected to go to one and only one multiplexer output at any given time, i.e., all outputs get connected to a unique and different input (also called unicast connections).



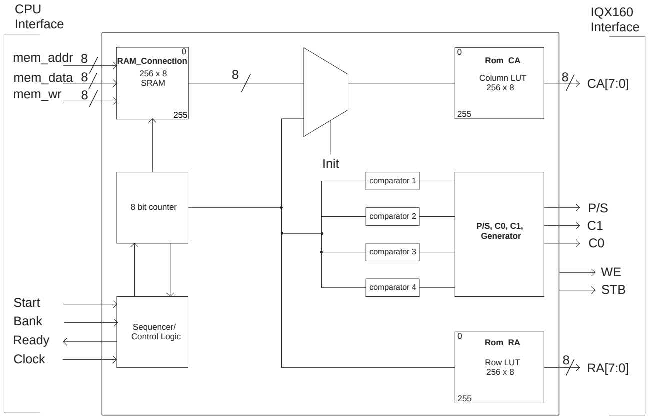
Bank Switching Using IQX Devices cont'd.

- Implemented as two 48x24 multiplexers
  - Duplicate inputs signals
  - Use Input Tristate (IT) with GT0) tristate control source
- RapidConfigure controller automatically configures I/O's at power_on

The Bank Switching multiplexer is logically implemented as two 48 x 24 multiplexers inside a single IQX160. The 48 signal input block is duplicated at the input side. This requires a total of 96 I/O Ports, with each incoming multiplexer input going to two I/O Ports on the IQX160. The multiplexer input signals in these two blocks are identified with prefixes A and B. For example, the first multiplexer input, called IN0, is duplicated as AIN0 and BIN0 and connected to two I/O Ports on the IQX160. All 160 I/O Ports on the IQX160 are identical and can be programmed as either input or output. The exact assignment of I/O Ports on the IQX160 for multiplexer inputs and outputs is quite critical and requires careful consideration, as explained in the application brief.

The programmable I/O Ports used for the multiplexer input block A (AIN0 through AIN47) are configured as "Input with Tristate Control (IT)", with GT0 as the tristate control source. The I/O Ports used for the multiplexer input block B (BIN0 through BIN47) are also configured as "Input with Tristate Control (IT), with the same tristate control signal GT0 but with inverted polarity. The I/O Ports used for the 24 multiplexer outputs are configured as "Output (OP)". If desired, these I/O Ports could also be configured as "Output with Tristate Control (OT)", with GT1 as the tristate control signal.

Bank Switching Using IQX Devices cont’d.



RapidConfigure Controller for Bank Switching Using IQX160

The external hardware for the RapidConfigure interface consists of a small FPGA to implement the control logic and two small 256 x 8 Look-Up Tables (LUTs) for mapping of multiplexer input or output numbers to IQX160 I/O ports, and to store initialization sequence data.

Column Look Up Table (ROM_CA)

This LUT (256 x 8) contains port mapping data for output ports. It maps user output ports (0..23) to IQX160 I/O ports used as output ports.

Row Look Up Table (ROM_RA)

This LUT (256 x 8) contains port mapping data for input ports. It maps user input ports (0..47) for Bank A and (48..95) for Bank B to IQX160 I/O ports used as input ports.

Connection Map SRAM: (RAM_Connection)

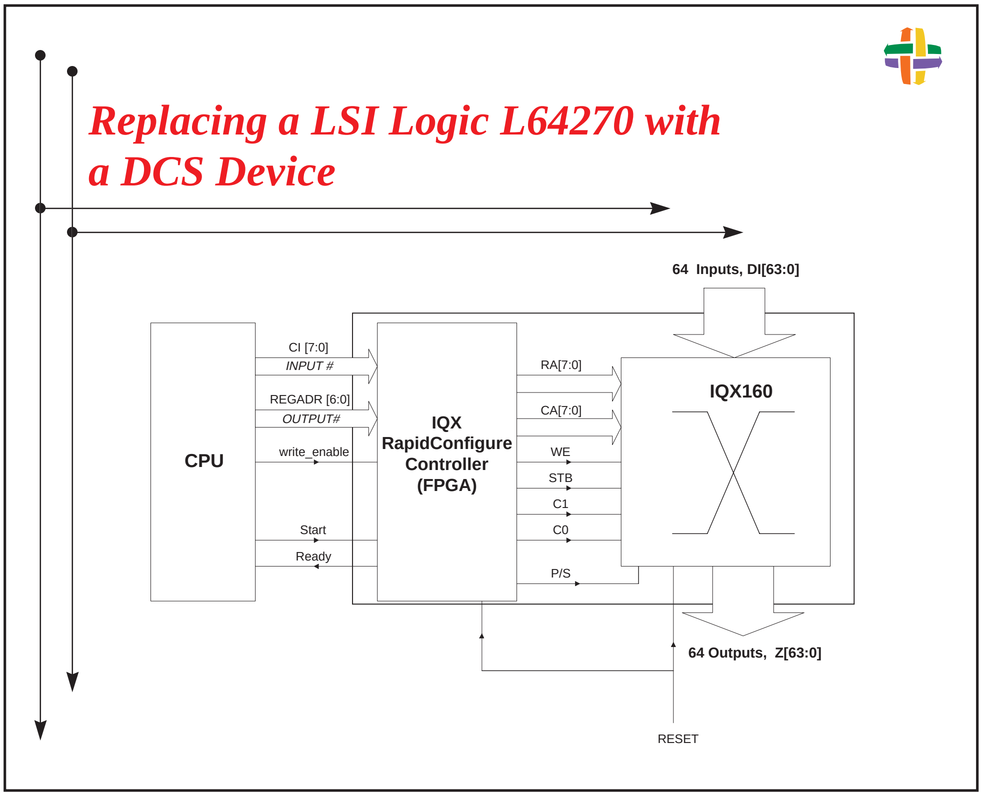
This SRAM (256 x 8) contains the connection map for both Bank A and Bank B. Data in address locations (0..47) contains the user output port number that connects to input ports (0..47) in Bank A. Data in address location (48..95) contain the user output port number that connects to input port (0..47) in Bank B.

Comparator (1, 2, 3, 4)

These comparators are used during initialization to generate correct values for P/S, C1 and C0 signals.

8-Bit Counter

The 8-Bit counter is used to generate addresses (132..235) for ROM_RA and ROM_CA during the initialization sequence.

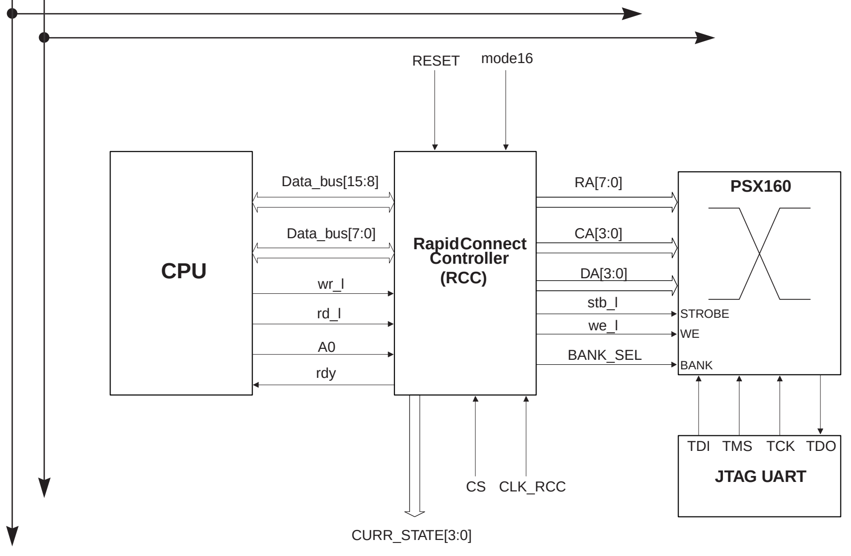


Replacing a LSI Logic L64270 with a DCS Device

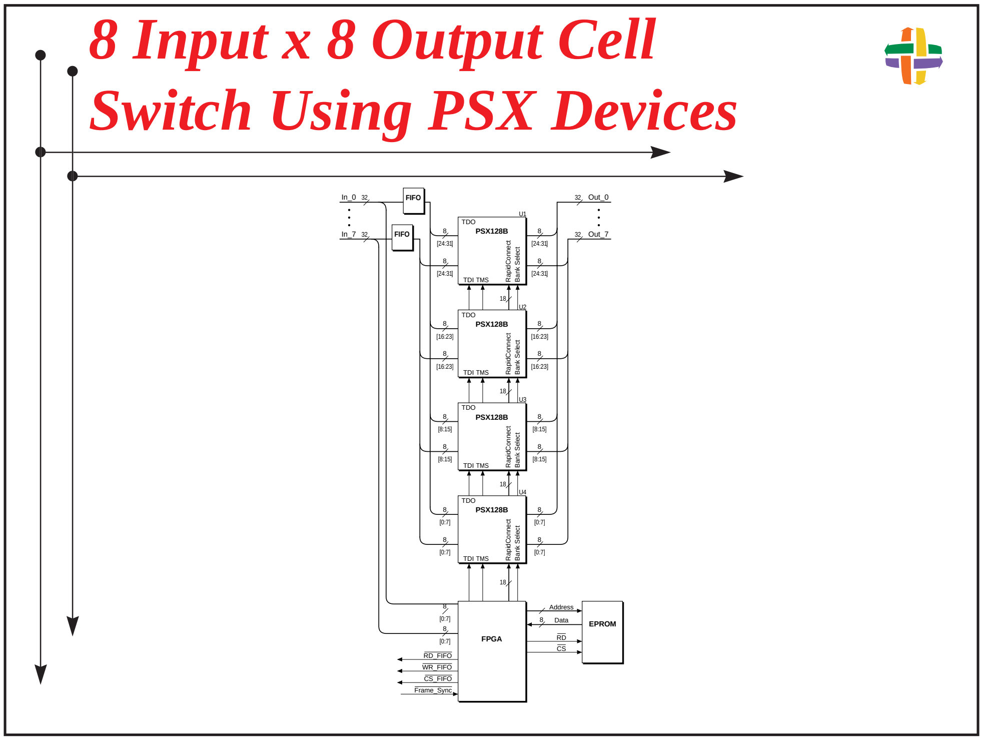
This application note is a complete design examples for replacing a LSI Logic L64270 with an I-Cube DCS device. This will be useful for all L64270 users wishing to upgrade their system design. This design examples uses an IQX160 to replace a L64270. The VHDL code presented for the switch controller assumes that the design is operating in flow-through mode. The VHDL code allows the FPGA that implements the switch controller to duplicate the functionality of the L64270 control interface to minimize the control system software changes the user needs to make.

Although the solutions presented are not 100% pin compatible with the LSI Logic L64270, the PCB changes are minimal. For those users who want to keep software compatibility the switch controller implemented via VHDL code for a FPGA allows the user to implement the LSI Logic L64270 interface-to-DCS device-interface, so maintaining software compatibility. Finally for next generation designs, which need larger switches, high performance, or greater flexibility, the DCS devices present an easy to use, ready made solution for user’s who are familiar with crossbar switches.

RapidConnect Controller for PSX Devices

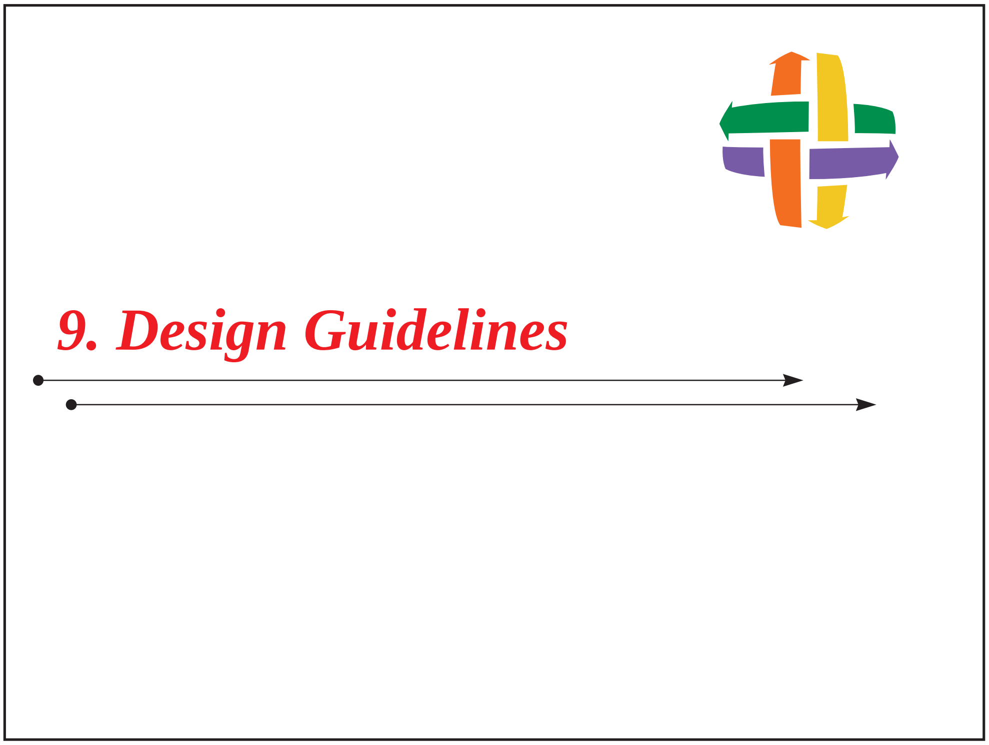


I-Cube's PSX devices are a family of bus-oriented switches designed for high performance, flexible switching applications. In order to achieve high performance switch changes a parallel interface called RapidConnect is used to address the switch matrix and make or break connections between groups of I/O ports (buses). The purpose of this application note is to describe how to design and implement a controller to simplify the RapidConnect interface for PSX devices and add a higher level functionality.



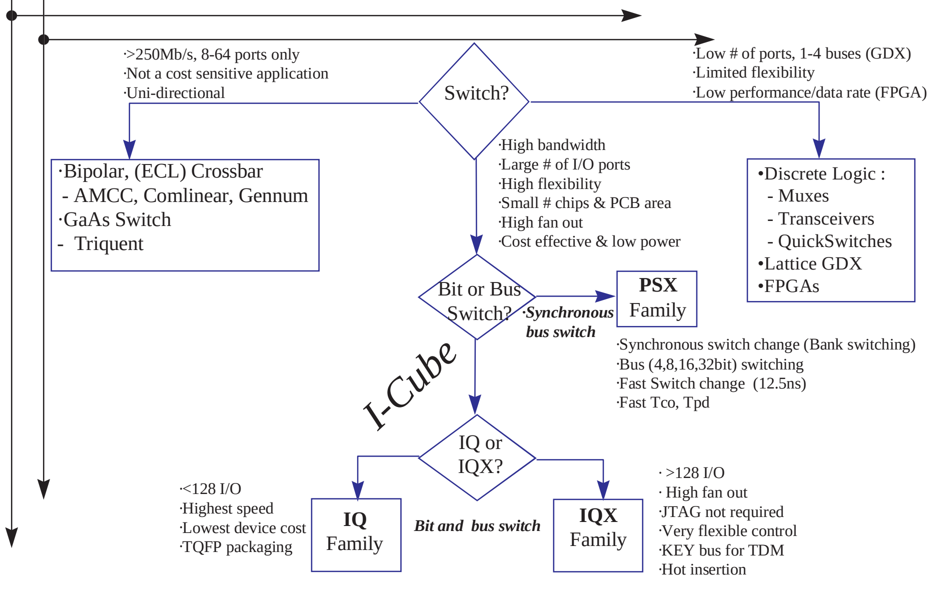
8 Input x 8 Output Cell Switch Using PSX Devices

This application note illustrates how to implement an 8 Input x 8 Output x 32-bit cell switch using 4 PSX128B, a small FPGA and a FIFO. The PSX device is used to switch data packets from the eight 32 bit-wide input channels to eight output channels. The FPGA is used for the control logic required to decode the connection map and generate the RapidConnect vectors for the PSX. The FIFO is used to buffer a single incoming data packet in order to pipeline configuration set up and data transfer.



9. Design Guidelines

Selecting a Switch



·>250Mb/s, 8-64 ports only
·Not a cost sensitive application
·Uni-directional

Switch?

·Low # of ports, 1-4 buses (GDX)
·Limited flexibility
·Low performance/data rate (FPGA)

·Bipolar, (ECL) Crossbar
- AMCC, Comlinear, Gennum
·GaAs Switch
- Triquent

·High bandwidth
·Large # of I/O ports
·High flexibility
·Small # chips & PCB area
·High fan out
·Cost effective & low power

•Discrete Logic :
- Muxes
- Transceivers
- QuickSwitches
•Lattice GDX
•FPGAs

Bit or Bus Switch?

·Synchronous bus switch

PSX Family

·Synchronous switch change (Bank switching)
·Bus (4,8,16,32bit) switching
·Fast Switch change (12.5ns)
·Fast Tco, Tpd

I-Cube

IQ or IQX?

·<128 I/O
·Highest speed
·Lowest device cost
·TQFP packaging

IQ Family

Bit and bus switch

IQX Family

· >128 I/O
· High fan out
·JTAG not required
·Very flexible control
·KEY bus for TDM
·Hot insertion

This flow chart is intended to give the user an easy selection guide on which DCS device family to use to implement a design. It can be used to select the preferred DCS device family (IQ, IQX, or PSX), and then SwitchSelector can be used to select the device(s) within the specific family.

•Use IQ32B-96 for low I/O, cost-sensitive designs

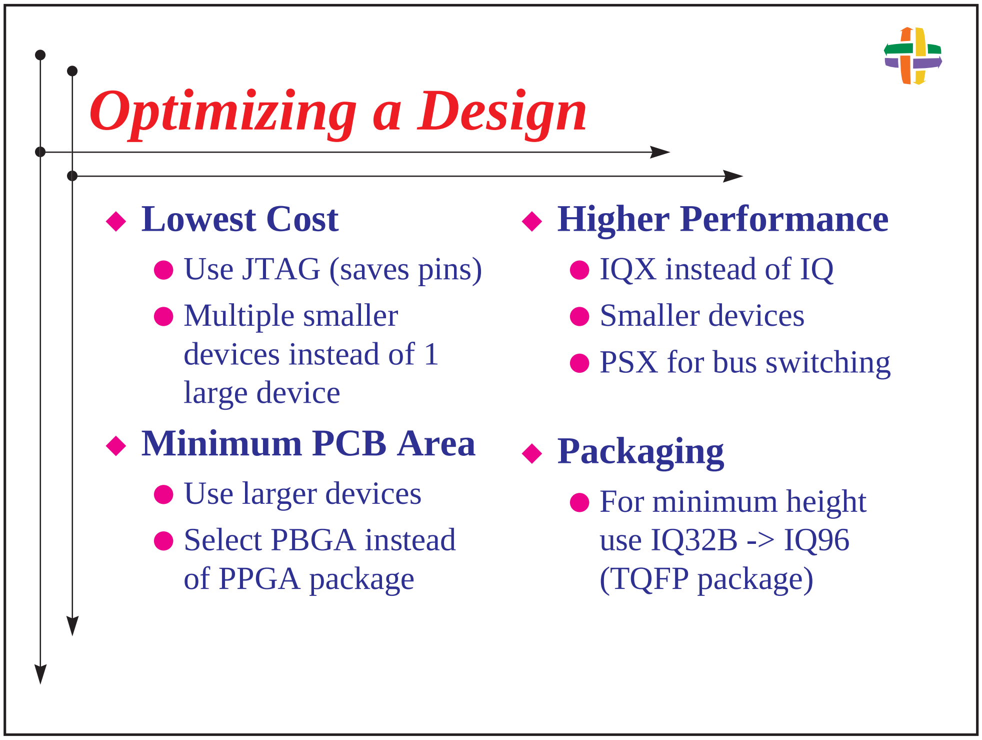
•Focus on IQX128B / 160 for large switch matrix & cost effective designs

–Remember 'bus switch' by 'bit slicing'

•Limit IQX320/240B for designs which need maximum flexibility or minimum PCB area

–IQX rather than IQ PBGA is preferred to PPGA

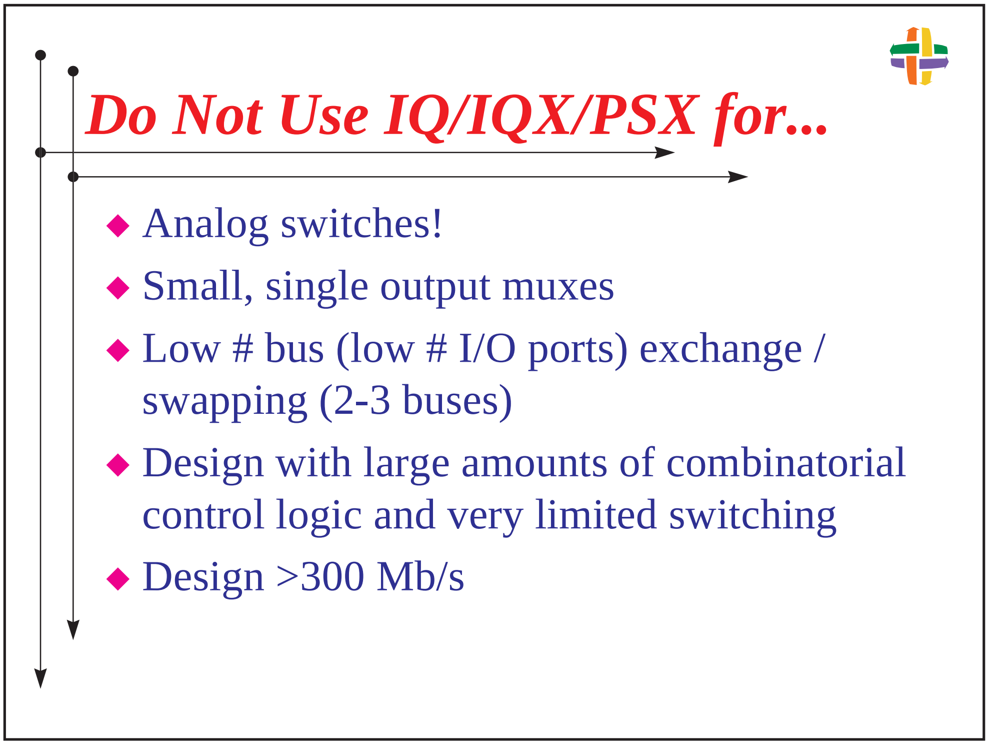
•Limit PSX for 4/8/16/32 bit buses which need fast synchronized (bank) switching



Optimizing a Design

- Lowest Cost
  - Use JTAG (saves pins)
  - Multiple smaller devices instead of 1 large device
- Minimum PCB Area
  - Use larger devices
  - Select PBGA instead of PPGA package
- Higher Performance
  - IQX instead of IQ
  - Smaller devices
  - PSX for bus switching
- Packaging
  - For minimum height use IQ32B -> IQ96 (TQFP package)

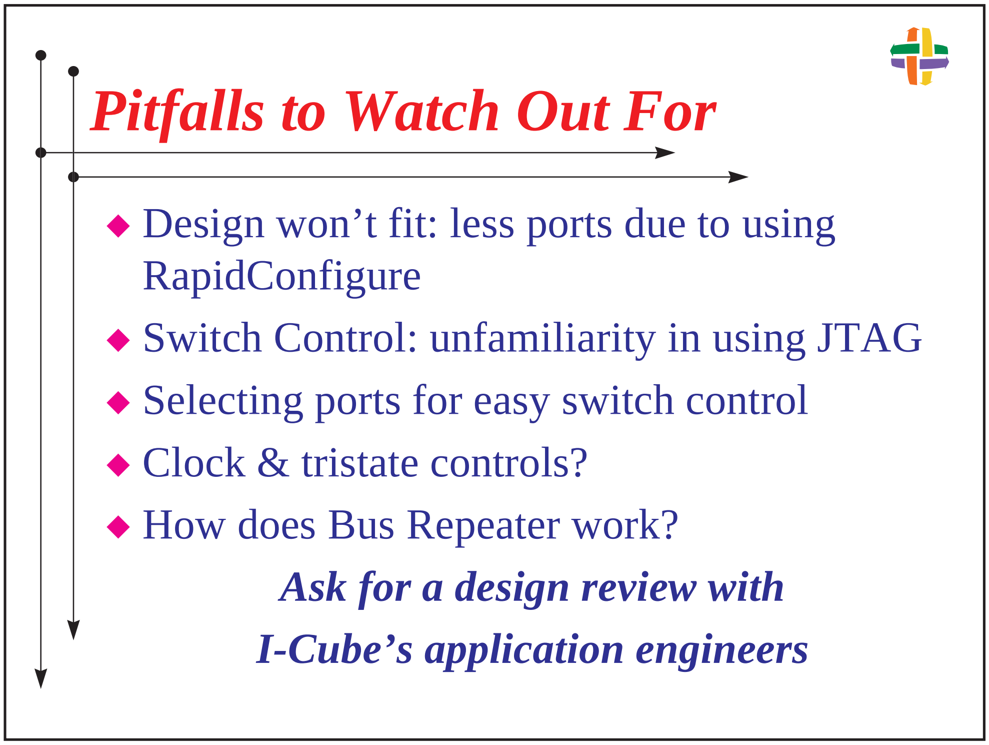
These are four areas the user may wish to optimize a design, with recommendations on which DCS device to select in order to meet the most important criteria.



Do Not Use IQ/IQX/PSX for...

- Analog switches!
- Small, single output muxes
- Low # bus (low # I/O ports) exchange / swapping (2-3 buses)
- Design with large amounts of combinatorial control logic and very limited switching
- Design >300 Mb/s

DCS devices are great for many applications, but there are some designs where other solutions can offer the user a “better” alternative.



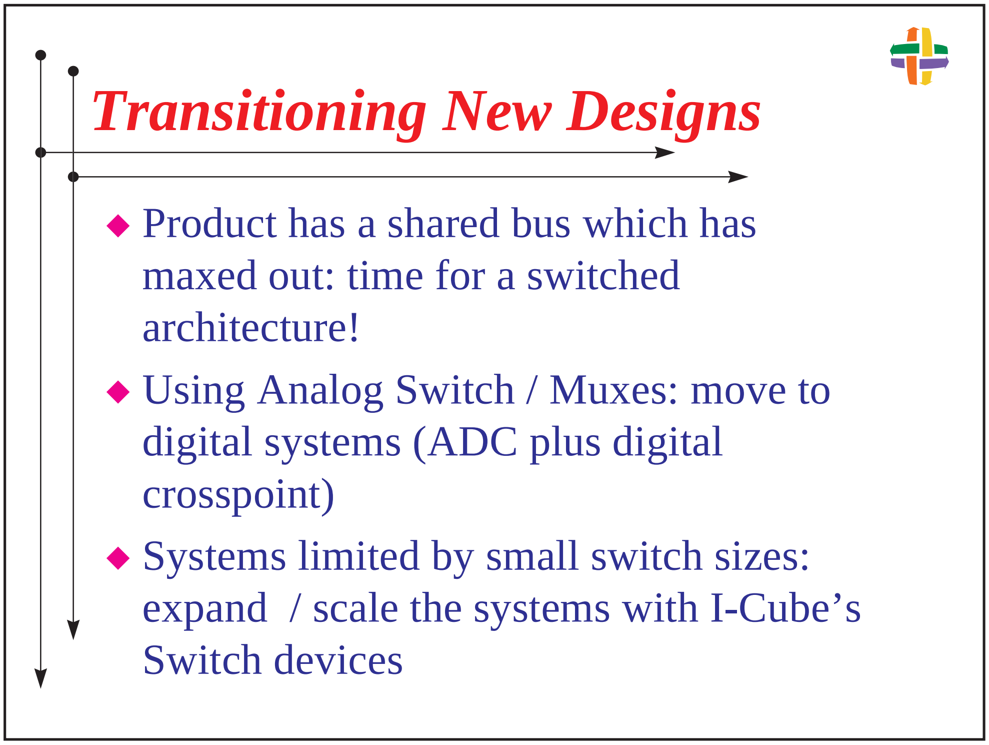
Pitfalls to Watch Out For

- ◆ Design won’t fit: less ports due to using RapidConfigure
- ◆ Switch Control: unfamiliarity in using JTAG
- ◆ Selecting ports for easy switch control
- ◆ Clock & tristate controls?
- ◆ How does Bus Repeater work?

Ask for a design review with I-Cube’s application engineers

I-Cube’s application engineers are available to help at all stages of the system design, from initial architecture discussions through to design implementation and prototype debugging.

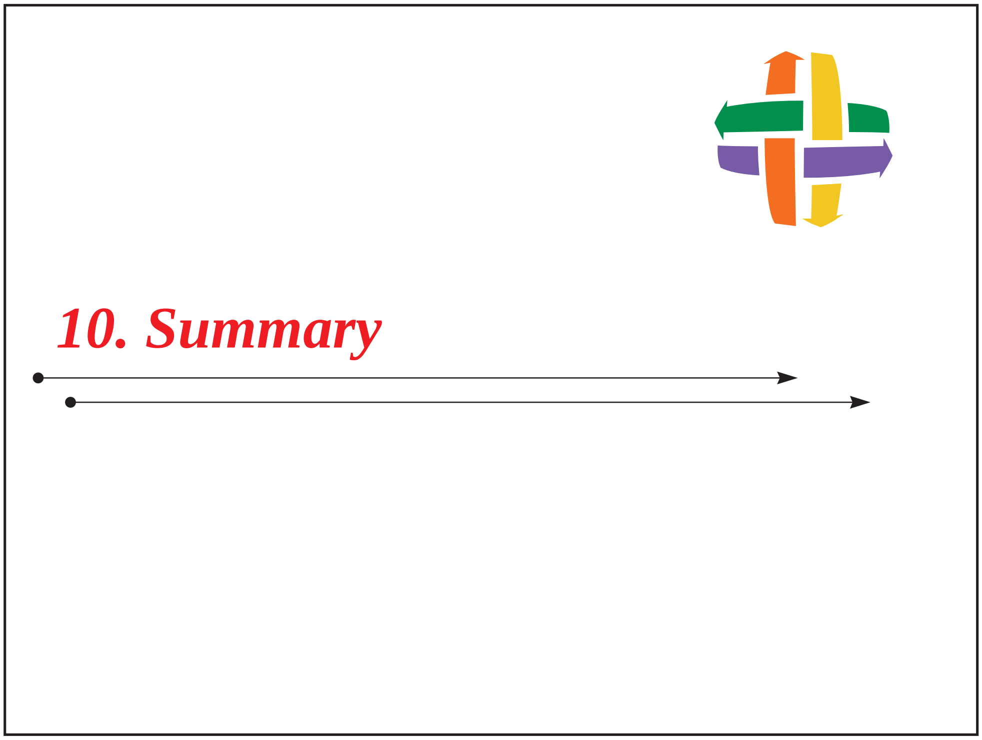
I-Cube encourages our customers to ask for a design review with our application engineers to help get your system working as quickly as possible so that you meet your project milestones and your time-to-market schedule.



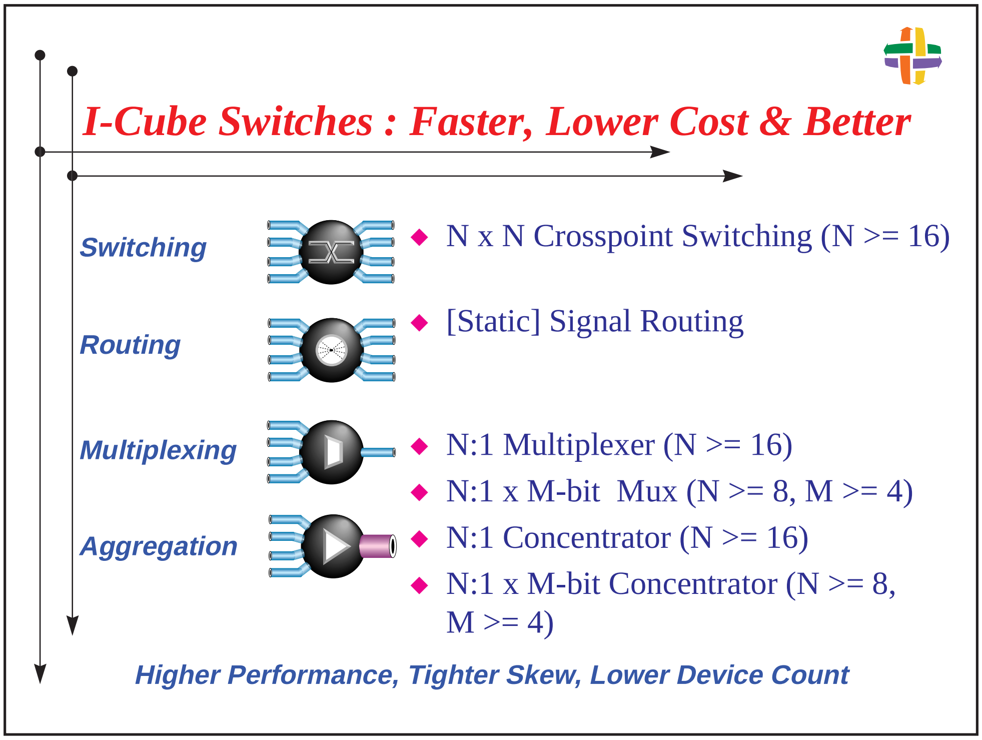
Transitioning New Designs

- Product has a shared bus which has maxed out: time for a switched architecture!
- Using Analog Switch / Muxes: move to digital systems (ADC plus digital crosspoint)
- Systems limited by small switch sizes: expand / scale the systems with I-Cube’s Switch devices

DCS devices and switching are key trends for future system designs requiring high bandwidth on demand. As current architectures “max out” many opportunities exist to evolve to switched architectures taking advantage of features offered by DCS devices.



10. Summary



I-Cube Switches : Faster, Lower Cost & Better

- N x N Crosspoint Switching (N >= 16)
- [Static] Signal Routing
- N:1 Multiplexer (N >= 16)
- N:1 x M-bit Mux (N >= 8, M >= 4)
- N:1 Concentrator (N >= 16)
- N:1 x M-bit Concentrator (N >= 8, M >= 4)

Higher Performance, Tighter Skew, Lower Device Count

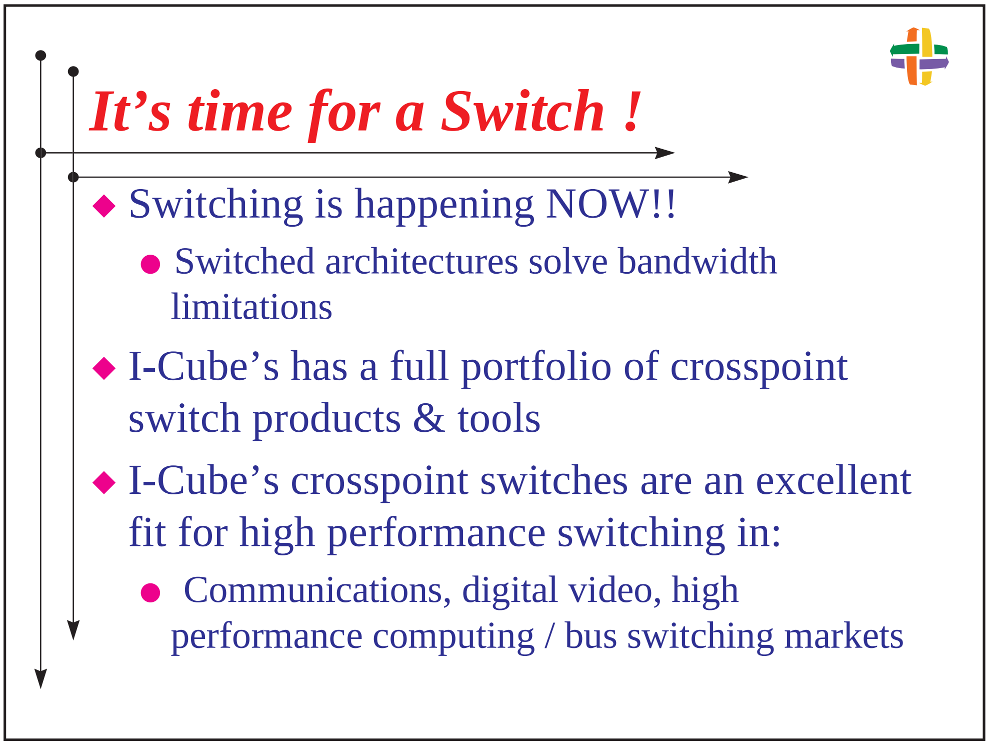
DCS devices are ideally suited to perform the following functions and offer a great deal of value to the system designer for the following applications.

Crosspoint switching applications, particularly those requiring a large switching element. Smaller switches can be built with FPGAs and even banks of TTL muxes; however, when you require switching among 16 or more inputs, or require data rates in 100 Mbs (50MHz clock) range, then the DCS devices (IQ and IQX) offer a good, cost effective solution.

DCS devices (IQX and PSX) can be used for constructing **multiplexers and demultiplexers** . Just like the crosspoint switches, small muxes can be built out of FPGAs or TTL SSI/MSI components; but when you require selecting between a large number of signals (16 and above), or when the data path is wide (8 or more inputs with 4-bit or wider data path), or when data rates are high (50 MHz clock), DCS devices are an excellent fit.

Data aggregation / concentration applications requiring a large number of incoming channels (16 or more) or wide channels, DCS devices are a good choice.

Static routing applications requiring routing among a large number of signals (16 or more), DCS devices (IQ and IQX) are the best choice. The higher the data rate, the better we look.



It’s time for a Switch !

- Switching is happening NOW!!
  - Switched architectures solve bandwidth limitations
- I-Cube’s has a full portfolio of crosspoint switch products & tools
- I-Cube’s crosspoint switches are an excellent fit for high performance switching in:
  - Communications, digital video, high performance computing / bus switching markets

Switching-technology is set to follow a rapid adoption curve; already innovators, especially in telecommunication and networking applications, have been using switching technology for many years. As component costs have decreased and ASIC technology have improved we have seen ***early adopters*** such as SGI, Sun, HP and IBM move to use switch-based technology.

Now with the availability of high performance standard products from I-Cube the ***mainstream*** users will be able to take advantage of the many benefits offered by switch-based architectures.

I-Cube Contact Information

- **Sales :**
  - Tel : 408 341 1888 press ‘3’
- **Marketing :**
  - Tel : 408 341 1888 press ‘4’
  - Email : *psid@icube.com*
- **Applications :**
  - Tel : 408 341 1888 press ’5’
  - Email *psid@icube.com*
- Web Site : **www.icube.com**

I-Cube, Inc.
2605 S. Winchester Blvd..
Campbell, CA 95008
Tel: (408) 341 1888
Fax: (408) 341 1899
Web site: www.icube.com

I-Cube®

Digital Crosspoint Switching Products

Quality Package

July 1998

I-Cube, Inc. •2605 S. Winchester Blvd. •Campbell, CA 95008
Tel (408) 341-1888 •Fax (408) 341-1899

Table Of Contents

1.0 Introduction

I-Cube's manufacturing strategy is to establish long term relationship with multiple world-class wafer foundries, die assembly, and product testing subcontractors. I-Cube's multiple sources strategy ensures the supply of high quality, high volume products with competitive cost and fast delivery.

I-Cube's product wafers are processed in sub micron 6" wafer fabs of UMC and TSMC, both are in Hsinchu, Taiwan. These facility are equipped with the state-of-the-art equipment capable of producing wafers with sub-half micron feature size. All I-Cube's Digital Crosspoint Switching (DCS) products available today are produced using advanced 0.8 micron and 0.6 micron CMOS single poly double metal silicon processing technologies. Starting wafers are lightly doped p-type silicon substrates. These technologies are volume production proven.

To ensure high quality, high volume integrated circuit packaging, I-Cube has all its products assembled by Amkor/Anam in Seoul and Philippine, and by Hana Technologies in Hong Kong. In addition to low cost Plastic Quad Flat Pack (PQFP) packages, I-Cube's products employ thermal and electrical enhanced packages, Thermally Enhanced Ball Grid Array from Prolinx, for applications with very high bandwidth data transfer.

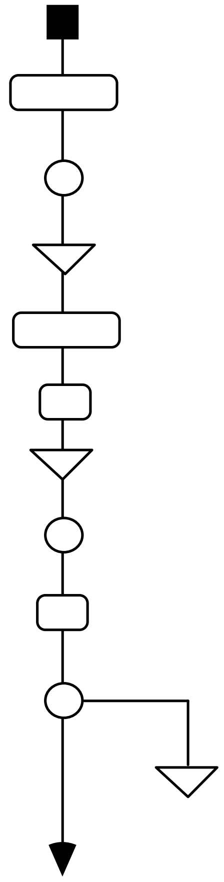
Today, I-Cube's wafer sort and packaged product testing are performed at DTS , which is located in Santa Clara, California. I-Cube's products are tested on high pin-counts, high frequency testers to assure the products' quality.

I-Cube is quality conscientious from product design through product testing. Quality control monitoring and quality assurance tests are performed by I-Cube and all its subcontractors at each facet of the production process.

This report summarizes the results of all quality and reliability tests for Digital Crosspoint Switching (DCS) products. Data for MQUAD and PPGA packages are also included in this report.

2.0 Production Flow

I-Cube Production Flow



Wafer Start / Laser Mark Wafers

Wafer Processing in Submicron Fabs
QC Monitoring at critical steps

QA Wafer PCM Test and
Visual Inspection

Wafers ship to I-Cube or Test Subcontractor

Incoming Visual Inspection
Wafer Sort / Full Functional Tests

Wafer Back Grinding

Wafers ship to Assembly Subcontractor

Assembly Incoming QA
Wafer Inspection

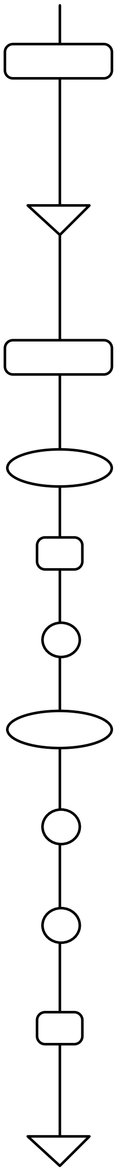
Die Saw

2nd OPT Inspection

(Ship Dice to Customers for Die Order)

(Continue)

I-Cube Production Flow (Continue)



Die Assembly (See Assembly Flow Chart for Details): Die Attach, Wire Bond, Die Coat, Lead Seal (or Plastic Mold), Bottom Mark, Lead Trim & Form, Top Mark, etc.

Packaged Products Tranfer to I-Cube or Test Subcontractor / Incoming QA Inspection

Product Test: Functional , DC, and AC Testing to Product Specifications
(Post Test QA Sampling)

Stage in Finish Goods Area

Top Mark / Speed Mark (Optional)

Final QA Sampling Test

Stage in Box Stocks Area

Coplanarity Test
(QA Sampling for Surface Mount Packages Only)

Final QA Visual Inspection

Unit Count and Packing
(Dry Packing for Surface Mount Packages Only)

Ship to Customers

3.0 Materials

3.1 Integrated Circuits

3.1.1 UMC / 0.8 micron CMOS SPDM technology
- p- silicon substrate
- N+ doped polysilicon gate electrode
- Metalization: Ti/TiN/Al(Si/Cu)

3.1.2 UMC / 0.6 micron CMOS SPDM technology
- p- silicon substrate
- N+ doped polysilicon gate electrode
- Metalization: Ti/TiN/Al(Si/Cu)

3.1.3 TSMC / 0.8 micron CMOS SPDM technology
- p- silicon substrate
- N+ doped WSix gate electrode
- Metalization: Ti/TiN/Al(Si/Cu)

3.1.4 TSMC / 0.6 micron CMOS SPDM technology
- p- silicon substrate
- N+ doped WSix gate electrode
- Metalization: Ti/TiN/Al(Si/Cu) with W plug

3.2 Package Construction

3.2.1 PQFP
- Lead frame: EFTEC 64T (Cu Alloy)
- Lead Plated: Sn/Pd = 85/15
- Bond wire: Gold 1.3 mil or 1.2 mil diameter
- Mold compound:

  SUMIKON EME-6300H
  Flammability - UL 94V-0
  Oxygen level: min. 28% (ASTM D2863-77)

3.2.2 Thermally Enhanced BGA
- Solder Ball: Sn/Pd = 63/37
- Bond wire: Gold 1.3 mil or 1.2 mil diameter
- Base (Copper Core):

  Photo-Imageable Dielectric
  Copper Slug Incorporated
  3-Layer substrate with Photo Imaged Via
  Cavity Down

4.0 Quality Assurance and Reliability

4.1 Qualification and Reliability Tests

I-Cube qualify new processing technologies, new packages, and new products based on the following tests and acceptance criteria:

Name of Test	Test Conditions	LTPD (1)	Sample Size	Acceptance
High Temperature Operating Life	1000 hours @ 125°C or equivalent Maximum rated operating voltage Dynamic burn-in	5	45	0
Temperature Humidity Bias	1000 hours @ 85 °C , 85% Relative Humidity Maximum rated operating voltage	5	45	0
Unbiased Pressure Pot	96 hours @ 121°C 15 PSI, Unbiased	5	45	0
Thermal Shock	200 cycles : -50 °C to +125 °C MIL-STD-883, Method 1011	10	22	0
Temperature Cycle (2)	200 cycles : -65 °C to +150 °C MIL-STD-883, Method 1010	10	22	0
ESD	MIL-STD-883, Method 3015		3	0
Latch-up	JEDEC Standard 17		3	0

(1)

Minimum Sample Size				Max. Rejects
LTPD=3	LTPD=5	LTPD=10	LTPD=15	allowed
76	45	22	15	0
129	77	38	25	1
176	105	52	34	2

(2) Condition B, -50 °C to +125 °C, is used for PPGA and PBGA packages.

4.2 Quality Assurance Monitoring Program

I-Cube is committed to offer high quality and highly reliable product to our customers and will perform the following tests quarterly, or as frequently as meaningful, to monitor the reliability and improve the quality of I-Cube's products:

Name of Test	Test Conditions	LTPD	Sample Size	Acceptance
High Temperature Operating Life	1000 hours @ 125°C or equivalent Maximum rated operating voltage Dynamic burn-in	5	45	0
Temperature Humidity Bias	1000 hours @ 85 °C , 85% Relative Humidity Maximum rated operating voltage	5	45	0
Pressure Pot	96 hours @ 121°C 15 PSI, Unbiased	5	45	0
Temperature Cycle	200 cycles : -65 °C to +150 °C MIL-STD-883, Method 1010	10	22	0

4.3 Test Results

4.3.1 HTOL (125°C or equivalent / Maximum rated operating voltage)

- IQ320 - PP391 (Covers IQ240B) / UMC 0.8µm CMOS

Lot #	Duration (Hrs)	Sample Size	Rejects
UFB1681	2740	49	0
F21559F	1168	48	0
FB3047B	1000	48	0
FC2735A	1000	48	0

- IQ160 - MQ208 (Covers IQ128B) / UMC 0.8µm CMOS

Lot #	Duration (Hrs)	Sample Size	Rejects
F21988A	1908	47	0
F21988B	1168	48	0
F62396H	1000	48	0
F62396J	1000	48	0
F12805E	1000	48	0
FA3046E	1000	48	0

- IQ64B - PQ100 (Covers IQ96) / TSMC 0.6µm CMOS

Lot #	Duration (Hrs)	Sample Size	Rejects
B606291B	2000	45	0

- IQ48 - PLCC48 (Covers IQ32B) / UMC 0.6µm CMOS

Lot #	Duration (Hrs)	Sample Size	Rejects
F52967D	4000	48	0

- PSX160 - PQ240 (Covers PSX128B, PSX96B) / TSMC 0.6µm CMOS

Lot #	Duration (Hrs)	Sample Size	Rejects
B61185D	1500	48	0
BA0970A	1500	48	0

- IQX160 - PQ208 (Covers IQX128B) / UMC 0.6µm CMOS

Lot #	Duration (Hrs)	Sample Size	Rejects
F82444C	1000	48	0
F81320B	1000	48	0
FB2345E	1000	48	0
FBC13A	4000	48	0
FBC14A	1000	48	0
F7M47C	3000	48	0

- IQX320 - PB416 (Covers IQX240B) / UMC 0.6µm CMOS

Lot #	Duration (Hrs)	Sample Size	Rejects
FA484B	3000	22	0

FC3121H	2000	24	0
F8FG2C	1000	27	0

4.3.2 Temperature Humidity Bias (85°C/85%RH, Maximum rated operating voltage)

- IQ320 - PP391

Lot #	Duration (Hrs)	Sample Size	Rejects
F21559G	1000	45	0

- IQ160 - PQ208

Lot #	Duration (Hrs)	Sample Size	Rejects
F21988	1000	45	0

- IQ160 - MQ208

Lot #	Duration (Hrs)	Sample Size	Rejects
F62396A	1000	45	0

- IQ320 - PB416

Lot #	Duration (Hrs)	Sample Size	Rejects
FB1681D	1000	45	0

- IQX160 - PQ208

Lot #	Duration (Hrs)	Sample Size	Rejects
F7MG3C	500	48	0

- IQX320 - PB416

Lot #	Duration (Hrs)	Sample Size	Rejects
FA484B	500	24	0

4.3.3 Preconditioning (JESD 22-A113 method)
Simulate condition of 84 Hrs between bake and bag

			Sample	
Product	Package	Lot #	Size	Rejects
IQ160	MQ208	F21988	99	0
IQ160	PQ208	F21988	117	0
IQ160	MQ208	F82546E	45	0
IQ160	PQ208	F12805E	45	0
IQ160	PQ208	F22643C	48	0
IQ240B	MQ304	FB1681C	89	0
IQ320	PP391	F21559D	22	0
IQ320	PP391	FB3047B	45	0
IQ320	PP391	F72743B	48	0

IQ320	PP391	FC2735A	22	0
IQ320	PB416	FB067B	24	0
IQ320	PB416	FB3047C	25	0
IQ320	PB416	FB1681D	56	0
IQ320	PB416	FC30139B	47	0
IQ320	PB416	FC30139C	24	0
IQ32B	TQ52	F52967E	48	0
IQ48	TQ80	F52967C	48	0
IQ64B	PQ100	B606291B	48	0
IQ64B	J84	B606291C	45	0
IQ64B	J84	B606291C	48	0
IQ96	TQ144	B606291F	48	0
IQ96	TQ144	B37095H	45	0
PSX160	PQ240	BA0970C	48	0
PSX160	PQ240	B61026C	48	0
PSX160	MQ240	B61026D	48	0
PSX160	PQ240	B61026D	58	0
PSX160	PQ240	B61026J	123	0
PSX160	PQ240	BA0970C	45	0
PSX96B	PQ160	B61026E	48	0
IQX128B	PQ184	F7M47D	48	0
IQX160	PQ208	FBC14A	48	0
IQX160	PQ208	F7MG3A	24	0
IQX160	PQ208	F7MG3C	48	0
IQX240B	PQ304	FA483C	24	0
IQX240B	MQ304	FA483A	24	0
IQX320	PB416	FA484B	24	0
IQX320	PB416	FC3121E	42	0
IQX320	PB416	FC3121G	48	0
IQX320	PB416	FC3121G	59	0
IQX320	PB416	F8FG2B	24	0

4.3.4 Temp Cycle

			Sample		
Product	Package	Lot #	Size	Rejects	Cycles
IQ160	MQ208	F21988	24	0	300
IQ160	PQ208	F21988	24	0	300
IQ240B	MQ304	FB1681C	24	0	150
IQ320	PP391	F21559D	22	0	300
IQ320	PB416	FB3047C	25	0	300
IQ64B	J84	B606291C	48	0	300
IQ64B	PQ80	B606291B	48	0	300
IQ96	TQ144	B606291F	48	0	200

IQ320	PB461	FB1681D	56	0	300
IQ320	PP391	F72743B	48	0	300
IQ32B	TQ52	F52967E	48	0	300
IQ48	TQ80	F52967C	48	0	300
PSX160	MQ240	B61026D	48	0	300
PSX160	PQ240	B61026C	48	0	300
PSX96B	PQ160	B61026E	36	0	300
PSX160	PQ240	B61026J	48	0	300
PSX160	MQ240	B61026D	48	0	300
IQ160	MQ208	F82546E	45	0	300
IQ160	PQ208	F12805E	45	0	300
IQ320	PP391	FC2735A	22	0	300
IQ64B	J84	B606291C	45	0	300
IQ96	TQ144	B37095H	45	0	300
PSX160	PQ240	BA0970C	45	0	300
IQX320	PB416	FC3121G	59	0	300
IQ320	PB461	F30139C	24	0	300
IQ160	PQ208	F22643C	48	0	300
IQ160	PQ208	F22643C	48	0	300
IQ320	PB416	FB067B	24	0	300
IQX160	PQ208	F7MG3C	48	0	300
IQX160	PQ208	F7MG3A	24	0	300
IQX240B	MQ304	FA483A	24	0	300
IQX240B	PQ304	FA483C	24	0	300
IQX320	PP391	F8FG2B	24	0	300
IQX320	PB416	FA484B	24	0	300
PSX160	PQ240	BA0970C	24	0	300

4.3.5 Thermal Shock

			Sample		
Product	Package	Lot #	Size	Rejects	Cycles
IQ160	MQ208	F62396A	22	0	400
IQ160	PQ208	F62396H	22	0	400
IQ160	PQ208	F22643C	48	0	400
IQ240B	MQ304	F62676B	22	0	400
IQ320	PB416	FB067B	24	0	400
IQ320	PP391	F21559G	22	0	400
IQ320	PB416	FB3047C	25	0	400
IQ320	PP391	F72743B	48	0	400
IQ320	PB416	F30139C	24	0	200
IQ32B	TQ52	F52967E	48	0	400
IQ48	TQ80	F52967C	48	0	400
IQ64B	J84	B606291C	48	0	400
IQ64B	PQ100	B606291B	48	0	400

IQ96	TQ144	B606291F	48	0	400
IQX160	PQ208	F7MG3A	24	0	200
IQX160	PQ208	F7MG3C	24	0	200
IQX240B	PQ304	FA483C	24	0	200
IQX240B	MQ304	FA483A	24	0	200
IQX320	PP391	F8FG2B	24	0	200
IQX320	PB416	FC3121G	59	0	200
PSX160	MQ240	B61026D	48	0	400
PSX160	PQ240	B61026C	48	0	400
PSX160	PQ240	B61026J	48	0	400
PSX160	MQ240	B61026D	48	0	400
PSX96B	PQ160	B61026E	36	0	400

4.3.6 Pressure Pot (121°C / 15 Psi)

			Sample		Duration
Product	Package	Lot #	Size	Rejects	(Hrs)
IQ160	PQ208	F22643C	48	0	168
IQ160	MQ208	F21988	45	0	168
IQ160	PQ208	F21988	45	0	168
IQ240	MQ304	FB1681C	45	0	168
IQ320	PB416	FB067B	24	0	168
IQ320	PP391	F21559D	45	0	168
IQ320	PB416	FB3047C	25	0	96
IQ320	PB416	FC30139B	47	0	168
IQ32B	TQ52	F52967E	48	0	168
IQ48	TQ80	F52967C	48	0	168
IQ64B	PQ100	B606291F	45	0	168
IQ64B	J84	B606291C	45	0	168
IQ96	TQ144	B606291F	45	0	168
IQX160	PQ208	F7MG3A	24	0	168
IQX160	PQ208	F7MG3C	48	0	168
IQX240B	PQ304	FA483C	24	0	168
IQX240B	MQ304	FA483A	24	0	168
IQX320	PP391	F8FG2B	24	0	168
IQX320	PB416	FC3121G	48	0	168
IQX320	PB416	FC3121E	42	8	168
PSX160	MQ240	B61026D	48	0	168
PSX160	PQ240	B61026C	48	0	168
PSX160	PQ240	B61026J	48	0	168
PSX160	MQ240	B61026D	48	0	168
PSX96B	PQ160	B61026E	48	0	168

4.3.7 ESD

• IQ160	Sample Size	Rejects
1. Human body model		
3000 Volts	3	0
2 Machine Model		
400 Volts	3	0

• IQ320	Sample Size	Rejects
1. Human body model		
2000 Volts	3	0

• PSX160	Sample Size	Rejects
1. Human body model		
2000 Volts	2	0

• IQX160	Sample Size	Rejects
1. Human body model		
2000 Volts	3	0

• IQX320	Sample Size	Rejects
1. Human body model		
2000 Volts	3	0

4.3.8 Latch-up

For IQ160, IQ320, IQ96, IQ48, PSX160, IQX160 and IQX320:

	Sample Size	Rejects
1. 350 mA @ 10V	3	0
2. - 400 mA @ -1.4V	3	0

4.4 FITS

Based on the HTOL data reported in 4.3.1, the predicted failure rates @ +55°C at a 60% confidence level are as follows:

	Device Hours @125°C	Derived FITS @+55°C
• IQ160	337740	57
• IQ320	286324	67
• IQ64B	90000	214
• IQ48	192000	101
• PSX160	144000	134
• IQX160	528000	37
• IQX320	141000	137

IQ160, IQ320, and their related products are processed in the same facility and the same 0.8 micron CMOS technology, the predicted IQ product family failure rate @ +55°C at a 60% confidence level are as follows:

	Device Hours @125°C	Derived FITS @+55°C
• IQ Product Family	624064	31

IQX160, IQX320, and their related products are processed in the same facility and the same 0.6 micron CMOS technology, the predicted IQX product family failure rate @ +55°C at a 60% confidence level are as follows:

	Device Hours @125°C	Derived FITS @+55°C
• IQX Product Family	669000	29

PSX160 and its related products are processed in the same facility and the same 0.6 micron CMOS technology, the predicted PSX product family failure rate @ +55°C at a 60% confidence level are as follows:

	Device Hours @125°C	Derived FITS @+55°C
• PSX Product Family	144000	134

I-Cube®

Design Support Tools

- VHDL*
- BSDL*
- Source Code*
- Spice Models*
- OrCAD Symbols*
- Verilog*

* These are not PDF files