EE178 Lecture Module 2

Eric Crabill

SJSU / Xilinx

Spring 2006

Lecture #4 Agenda

- Survey of implementation technologies.

Implementation Technologies

- Small scale and medium scale integration.
  - Up to about 200 gates per device
  - Most common is 74xx type devices
    - Gates, flip flops, latches.
    - Decoders, registers, counters, and other functional building blocks.

Implementation Technologies

- Large scale integration.
  - Ranging from 200 to 200,000 gates per device.
  - Small memories, programmable logic devices, custom designs.
- Very large scale integration.
  - Above 200,000 gates per device.
  - Often “gate count” is replaced by transistor count because these large designs have integrated memories, etc.

Implementation Technologies

- Survey of small and medium scale components by browsing data books.
  - Different functional classes.
  - Generally used as “glue” logic now, to help interface larger scale components.
  - Back in the day, large designs were done using this technology.

Implementation Technologies

- Advantages of small and medium scale, particularly with regard to 74xx stuff.
  - Easy to understand functions.
  - Exceptional signal visibility.
- Disadvantages.
  - Low logic density means big boards or small designs only.
  - Higher power consumption.
  - Cost per function, failure concerns.

Implementation Technologies

- Survey of large scale components, for logic design, particularly programmable logic devices in this density.
  - Many different flavors of devices; most draw on basic device types.
    - ROM, PLA, PAL = PLDs.
    - CPLDs
  - Can be used as glue logic but have enough available logic to implement significant designs in larger parts.

Implementation Technologies

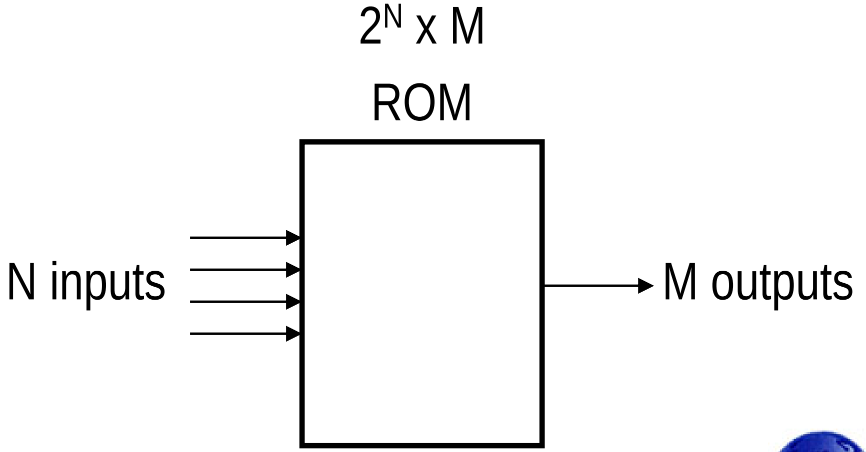
- Advantages of large scale integration.
  - Higher logic density means smaller boards or larger designs.
  - Many devices can be programmed and reprogrammed, saving expense when changes are made.
- Disadvantages.
  - Need to learn how to use and program.
  - Signal visibility is reduced.

Implementation Technologies

- What is a ROM? How can I use it?
- What is a PLA? How can I use it?
- What is a PAL? How can I use it?
- How are all these things related?
- What, then, is a CPLD?

Implementation Technologies

- A ROM is a SOP logic device with a fixed AND array and a programmable OR array.
- You can implement M functions of N inputs in this ROM.

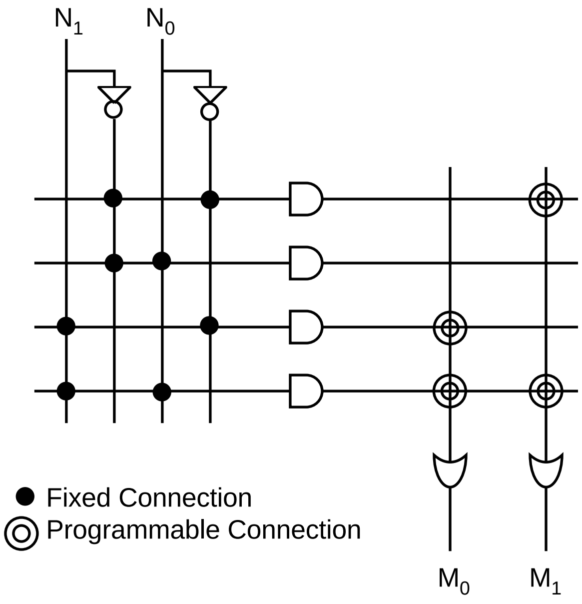


Implementation Technologies

- You basically specify a truth table of the functions when you program the ROM.
- There is no advantage to simplifying the function when you are using a ROM since you need to specify the entire list of minterms anyway…

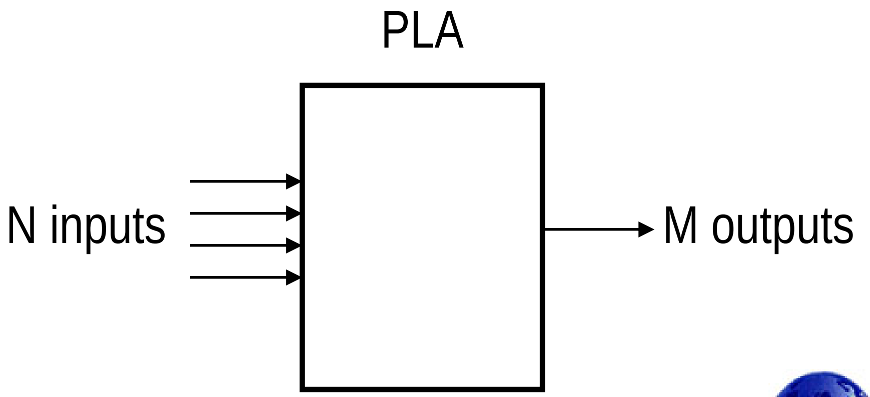
Implementation Technologies

- ROM of 2^N by M; N = 2, M=2
- M0 = N1•N0 + N1•N0’
- M1 = N1•N0 + N1’ •N0’



Implementation Technologies

- A PLA is a SOP logic device with a programmable AND array (fewer pt's than a ROM) and a programmable OR array.
- You can implement functions using the available minterms, which may be shared between functions.

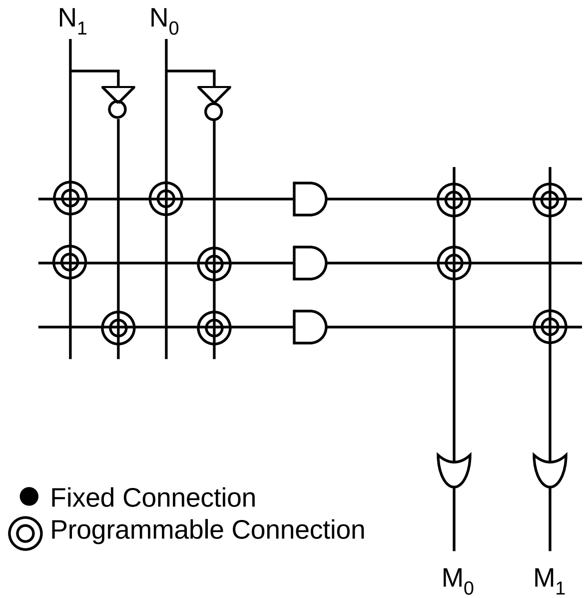


Implementation Technologies

- You have to reduce your design to a sum of products which will hopefully be realizable with the available minterms.
- Computer aided design tools are available to do optimization for product term sharing.

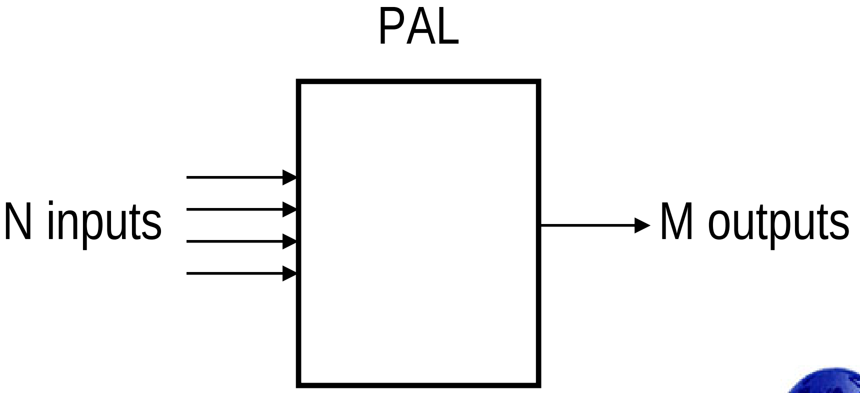
Implementation Technologies

- PLA of N inputs and M out; N = 2, M=2
- M0 = N1•N0 + N1•N0’
- M1 = N1•N0 + N1’ •N0’



Implementation Technologies

- A PAL is a SOP logic device with a programmable AND array and a fixed OR array.
- You can implement functions using the available minterms for each output function (no pt sharing).

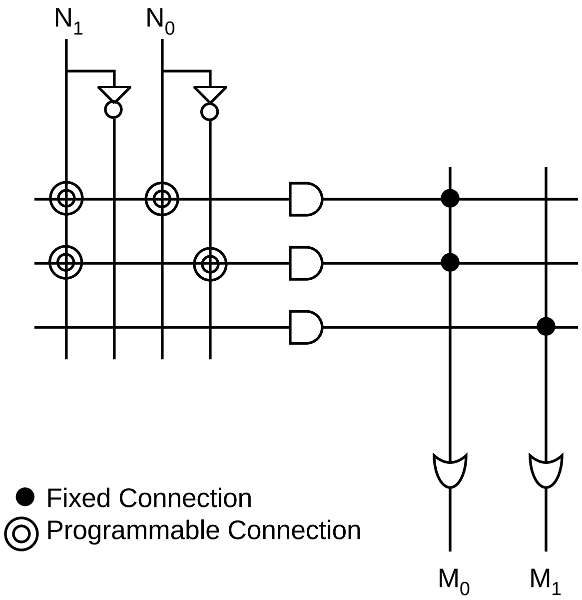


Implementation Technologies

- Again, the design has to be reduced if possible.
- No product term sharing, and note that in real devices, each output function may have access to a different number of product terms.

Implementation Technologies

- PAL of N inputs and M out; N = 2, M=2
- M0 = N1•N0 + N1•N0’
- M1 = N1•N0 + N1’ •N0’ insufficient minterms



Implementation Technologies

- A CPLD is a complex programmable logic device that essentially consists of a number of programmable logic blocks (such as a PLA, PAL, and less commonly, ROM) connected by a programmable interconnect array.
- Why has CPLD density stagnated?

Lecture #5 Agenda

- Survey of implementation technologies.

Implementation Technologies

- Full Custom Logic.
- Standard Cell Design.
- Gate Array Design.
- Field Programmable Logic.

Implementation Technologies

- Full Custom Logic.
  - Each primitive logic function or transistor is manually designed and optimized.
  - Most compact chip design, highest possible speed, lowest power consumption.
  - Non recurring engineering cost (NRE) is the highest for obvious reasons.
  - Rarely used today due to high engineering cost and low productivity; polygon pushing.

Implementation Technologies

- Standard Cell Design.
  - Predefined logic blocks (a la 74xx style) are made available to the designer in a cell library; the design is built with these.
  - Done with schematic capture or HDL.
  - Automated tools place and route the cells.
  - Cells are often standard dimensions to facilitate automated place and route.
  - Substantially shorter design time than custom.

Implementation Technologies

- Gate Array Design.
  - Full custom and standard cell require custom masks to produce wafers (read as “expensive”).
  - Instead, create base wafers using common masks; base wafers have an “array” of gates which are not committed and not wired.
  - Designers specify connectivity and the top metal masks are created to connect the gates on the base wafers.
  - Low wafer cost, fast turnaround, area penalty.

Implementation Technologies

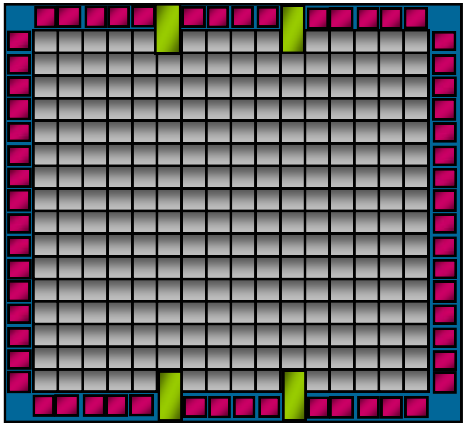
- Field Programmable Logic.
  - We have already discussed several types of programmable logic.
    - ROM, PLA, PAL.
    - CPLD.
  - The other main type of programmable logic is the field programmable gate array, or “FPGA”.

Field Programmable Gate Arrays

- FPGA devices are an improvement in gate array technology which offer improved time to market and reduced prototyping cost.
- Types of FPGA devices:
  - Non volatile, one time programmable (anti-fuse).
  - Non volatile, re-programmable (flash).
  - Volatile (sram).
- An FPGA is much more than an array of gates...

Field Programmable Gate Arrays

- What is in the array? All sorts of stuff…
  - I/O Cells.
  - Logic Cells.
  - Memories.
  - Microprocessors.
  - Clock Management.
  - High Speed I/O Transceivers.
  - Programmable routing.



XILINX

Field Programmable Gate Arrays

- The programmable routing is of particular significance because this is the main improvement over a standard gate array.
- An FPGA is really some programmable logic with a whole bunch of programmable wires!!!
- Various array sizes are available from the vendor.

Field Programmable Gate Arrays

- We will discuss the architectural details of the Xilinx Spartan-3 family of FPGAs in this class.
- You should be aware that Xilinx has other architectures, and that other companies have competing architectures.

Lecture #6 Agenda

- Spartan-3 FPGA family overview.
- Spartan-3 FPGA architecture detail part one.
- Technical information and diagrams reproduced with permission from Xilinx.

Xilinx Spartan-3 Family

- The Spartan product is a cost reduced, high volume FPGA. Most Spartan devices are a close relative to another Xilinx product.
- There are several Spartan FPGA families:
  - Spartan-II, Spartan-IIE (similar to Virtex).
  - Spartan-3, Spartan-3E (similar to Virtex-4).

Xilinx Spartan-3 Family

- EE178 currently uses the Spartan-3 FPGA family on a prototyping platform from Xilinx / Digilent.
  - High volume, 1.2 volt FPGA devices.
  - Pinout compatibility between devices.
  - On-chip memories and clock management.
  - Up to 5,000,000 system gates.

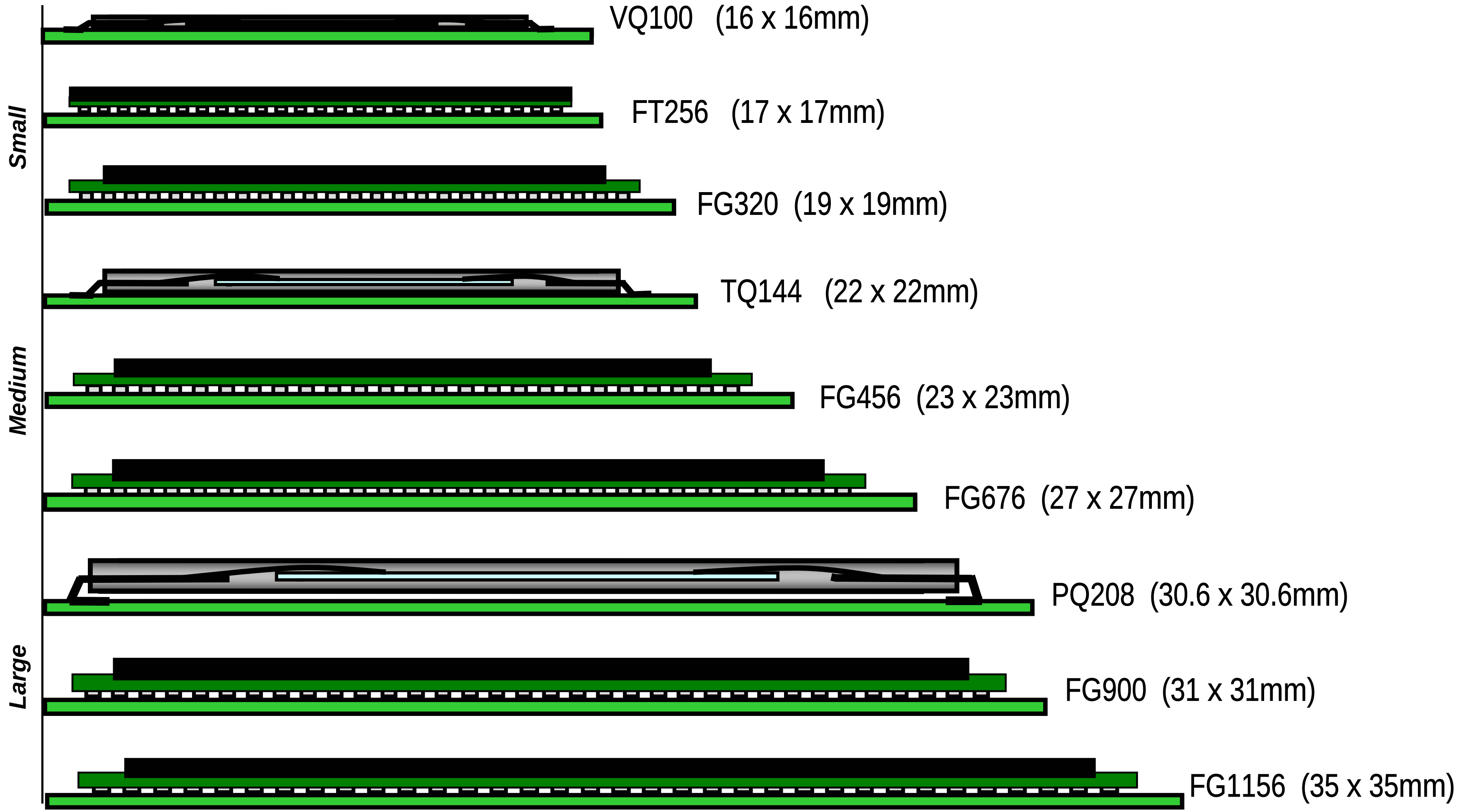
Spartan-3 Product Matrix

100X Density Range

Device	XC3S50	XC3S200	XC3S400	XC3S1000	XC3S1500	XC3S2000	XC3S4000	XC3S5000
System Gates	50K	200K	400K	1000K	1500K	2000K	4000K	5000K
Logic Cells	1,728	4,320	8,064	17,280	29,952	46,080	62,208	74,880
Dedicated Multipliers	4	12	16	24	32	40	96	104
Block RAM Blocks	4	12	16	24	32	40	96	104
Block RAM Bits	72K	216K	288K	432K	576K	720K	1,728K	1,872K
Distributed RAM Bits	12K	30K	56K	120K	208K	320K	432K	520K
DCMs	2	4	4	4	4	4	4	4
I/O Standards	24	24	24	24	24	24	24	24
Max Single Ended I/O	124	173	264	391	487	565	712	784

50,000 to 5,000,000 System Gates

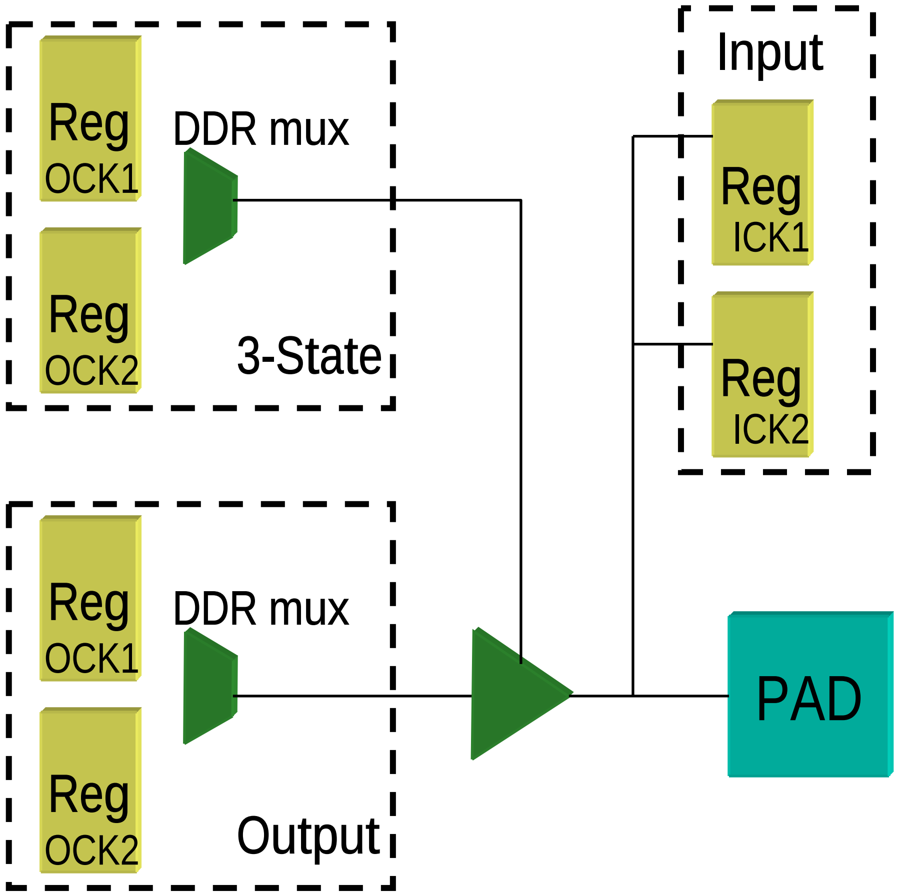
Choice of Packages



Xilinx Spartan-3 Family

- Programmable Input Output Blocks (IOB).
- Clock Management Blocks (DCM).
- Configurable Logic Blocks (CLB).
- Flexible Synchronous Memory (BlockRAM).
- A variety of programmable routing resources.

IOB Element

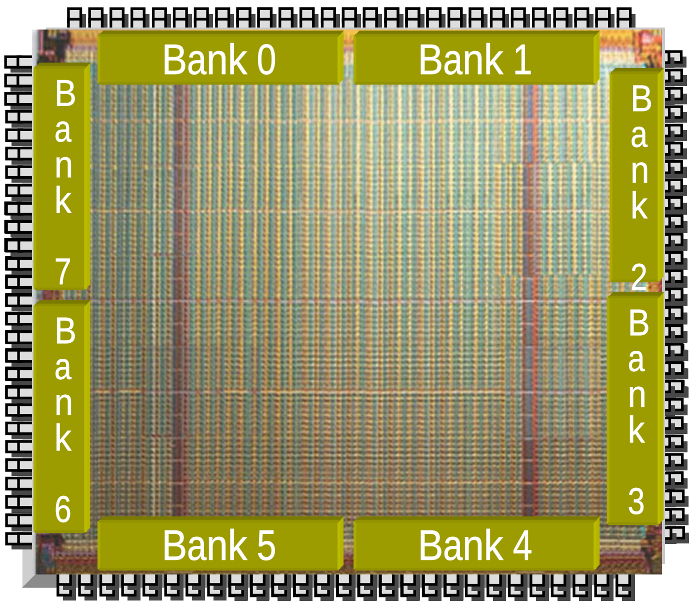


- Input path.
  - Two DDR registers.
- Output path.
  - Two DDR registers.
  - Two 3-state DDR registers.
- Separate clocks.
- Shared set and reset.
  - Separated sync/async.
  - Separated set/rst attribute per register.

I/O Block Advantages

- Independent registers.
  - Fast bus operations.
  - Interface to high-speed memory like ZBT and QDR.
  - Increase system performance with fast Tsu and Tco.
- Lower ground bounce with slew rate control.
- Zero hold-time for registered input signals using programmable input delay.
- Lower power consumption with keeper circuit.

Comprehensive Connectivity



- Single ended and differential.
  - 784 single-ended, 344 differential pairs.
  - 622 Mb/sec LVDS.
  - 24 I/O standards, 8 flexible I/O banks.
  - PCI 32/33 and 64/33 support.
  - Eliminate costly bus transceivers.
- 3.3V, 2.5V, 1.8V, 1.5V, 1.2V

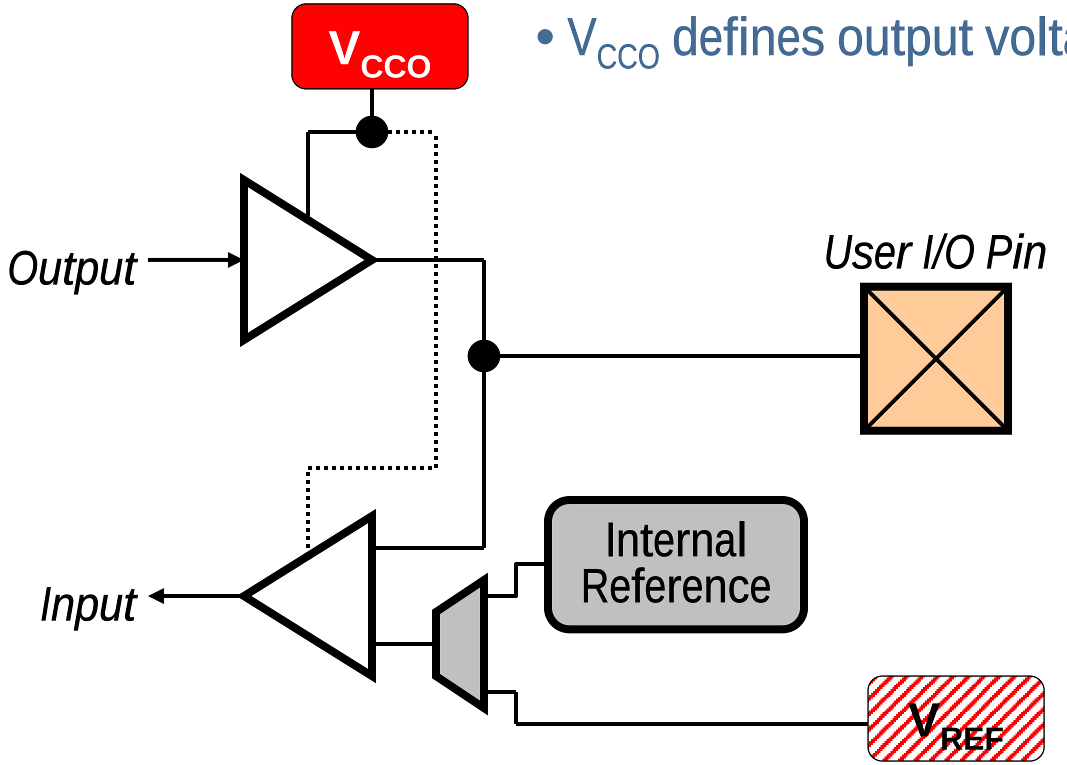
Chip-to-Chip Interfacing:	LVDS	LVCMOS	LVTTL	
Backplane Interfacing:	GTL	GTL+	PCI	BLVDS
High-speed Memory Interfacing:	HSTL	SSTL		

SelectIO

	Standard	Output V_{CCO}	Input V_{REF}
Single ended	LVTTL	3.3V	--
	LVCMOS33	3.3V	--
	LVCMOS25	2.5V	--
	LVCMOS18	1.8V	--
	LVCMOS15	1.5V	--
	LVCMOS12	1.2V	--
	PCI 32/64 bit 33MHz	3.3V	--
	SSTL2 Class I	2.5V	1.25V
	SSTL2 Class II	2.5V	1.25V
	SSTL18 Class I	1.8V	0.9V
	HSTL Class I	1.5V	0.75V
	HSTL Class III	1.5V	0.9V
	HSTL18 Class I	1.8V	0.9V
	HSTL18 Class II	1.8V	0.9V
	HSTL18 Class III	1.8V	1.1V
	GTL	--	0.8V
	GTL+	--	1.0V
Differential	LVDS2.5	2.5V	--
	Bus LVDS2.5	2.5V	--
	Ultra LVDS2.5	2.5V	--
	LVDS_ext2.5	2.5V	--
	RSDS	2.5V	--
	LDT2.5	2.5V	--

- More standards for system integration.
- Differential standards.
  - Higher I/O performance.
  - Lower power, lower cost.

SelectIO Standards



- V_{CCO} defines output voltage

- V_{REF} defines input threshold reference voltage
- Available as user I/O when using internal reference

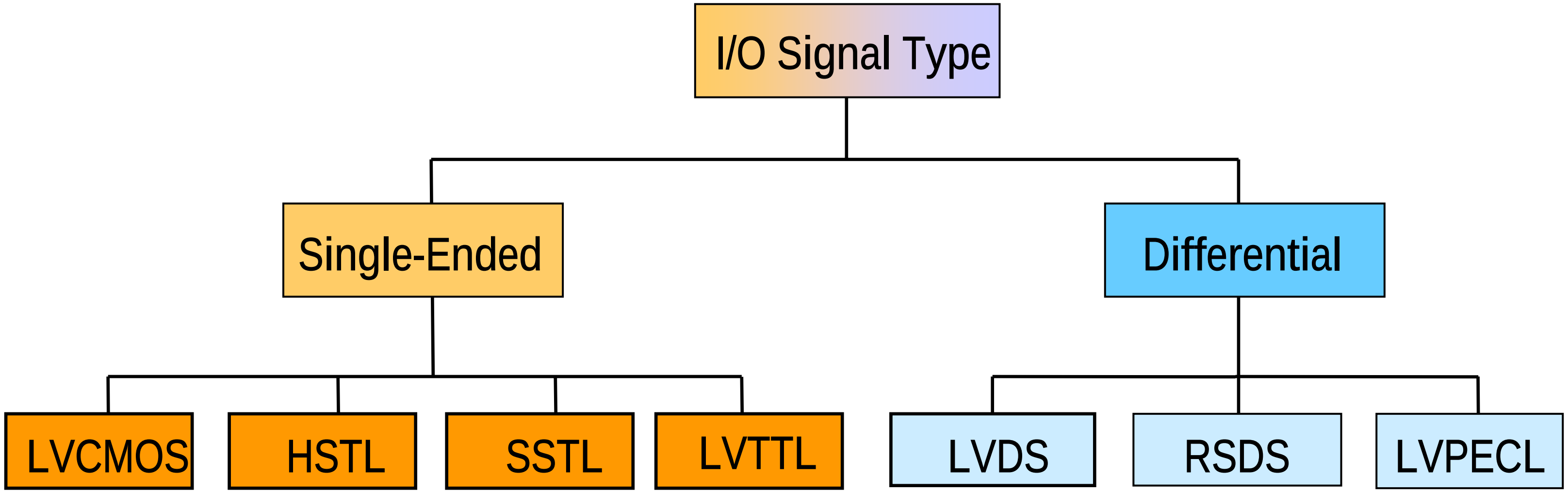
SelectIO Output Banks

- Each bank has an output driver voltage (V_{CCO}).
  - Shared among all I/Os in that bank.
  - All I/O in a bank must use the same voltage source.
  - All V_{CCO} pins in a bank must be the same voltage.
- Only one V_{cco} voltage for TQ144 per side.
- Outputs not requiring V_{CCO} fit in the bank.

SelectIO Input Banks

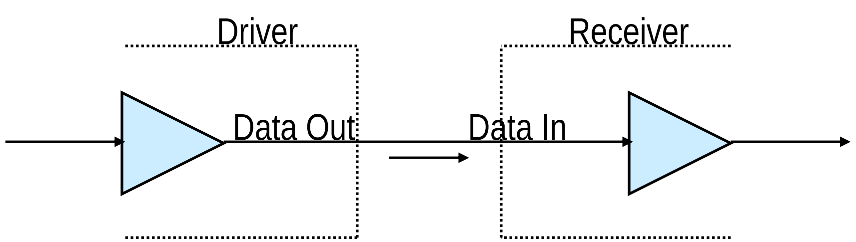
- Each bank has an input reference voltage (V_{REF}).
  - I/O in a bank must use the same reference voltage.
  - V_{REF} pins in a bank must be tied to the same voltage.
- Inputs not requiring a V_{REF} fit in the bank.
- V_{REF} pins in a bank available as additional I/O if I/O type does not require V_{REF}.

I/O Signal Types

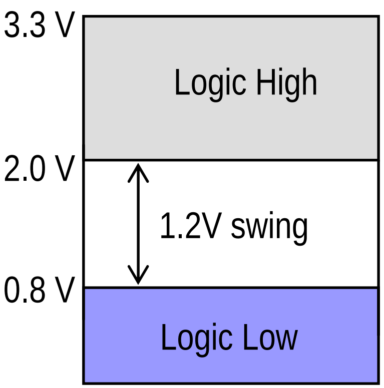


Single Ended I/O

- Traditional means of data transfer.
- Data is carried on a single line.
- Large voltage swing between logic levels.



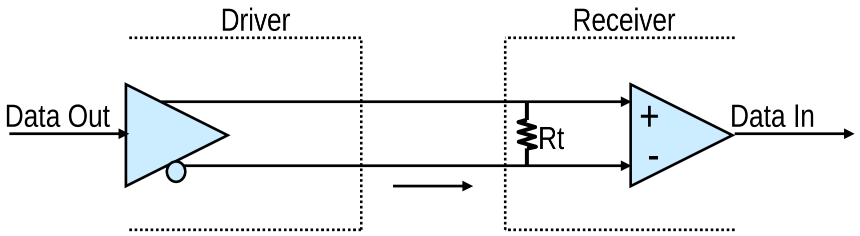
Single ended data transfer



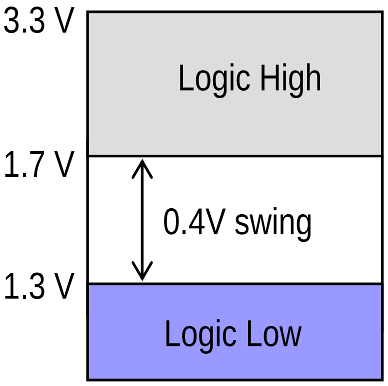
LVTTL input levels

Differential I/O

- One data bit is carried through two signal lines.
- Voltage difference determines logic level.
- Small voltage swing between logic levels.



Differential data transfer

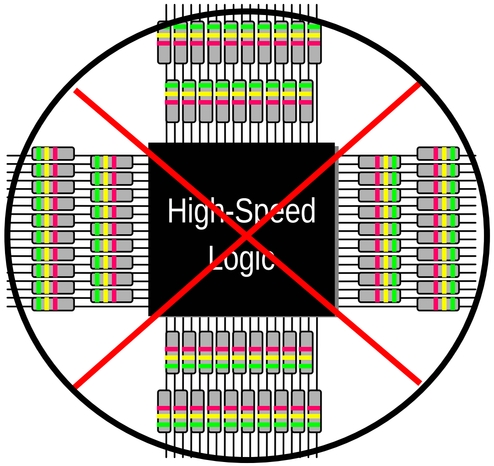


LVDS input levels

Differential I/O Benefits

- Small voltage swing between pairs.
  - Reduced emissions.
  - High performance per pin pair.
  - Reduced power consumption.
  - Improved noise rejection.
- Significant cost savings.
  - Fewer pins, board layers, board traces.
  - Smaller connectors.

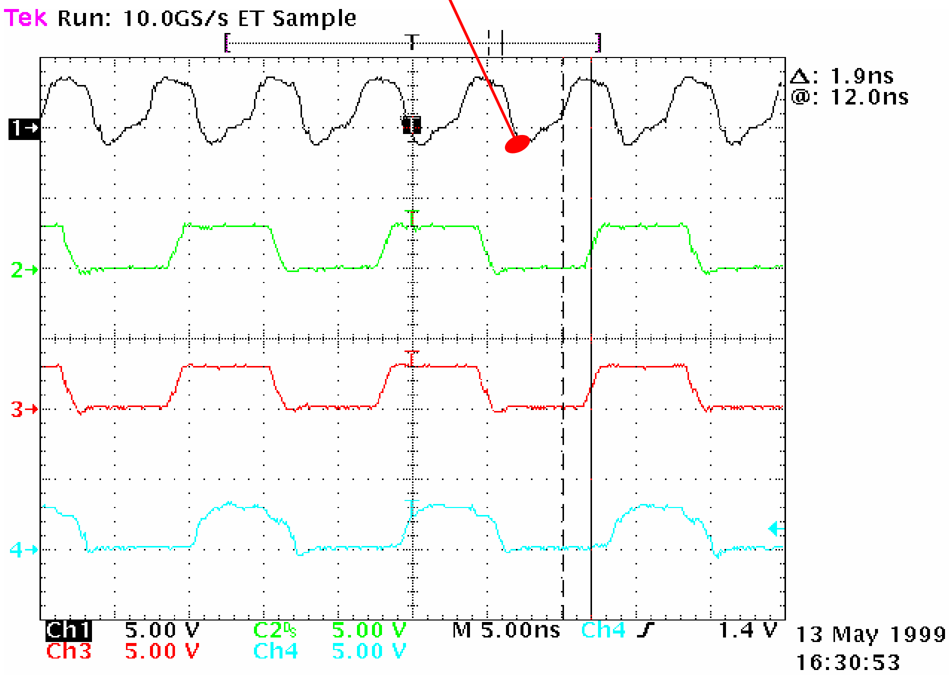
Digitally Controlled Impedance Drivers



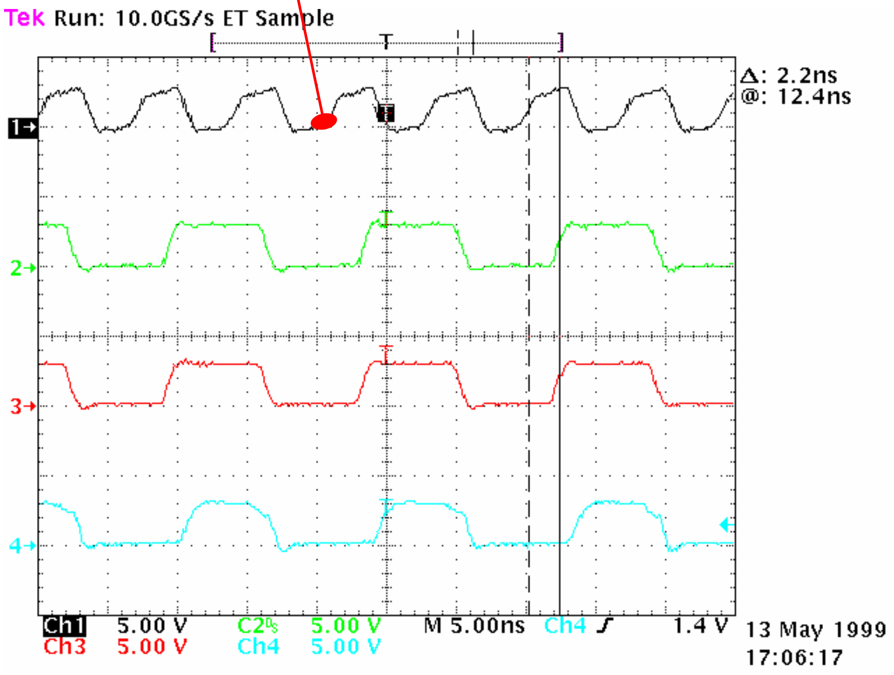
- On-chip I/O termination.
- Reduce total board cost.
  - Eliminate termination resistors.
  - Easier layout and fewer layers.
- Increases system reliability.
  - Greatly reduces component count.
  - Lower chance of failures.
- Elimination of stub reflection noise.
  - No traces between termination resistor and package pins.

Signal Integrity Adjustment

Initial Design: LVTTL_F16 (Fast slew, 16 mA)
Driver impedance too low – Undershoot!



Final Design: LVTTL_F8 (Fast slew, 8 mA)
Driver impedance ~50Ω -- No Undershoot



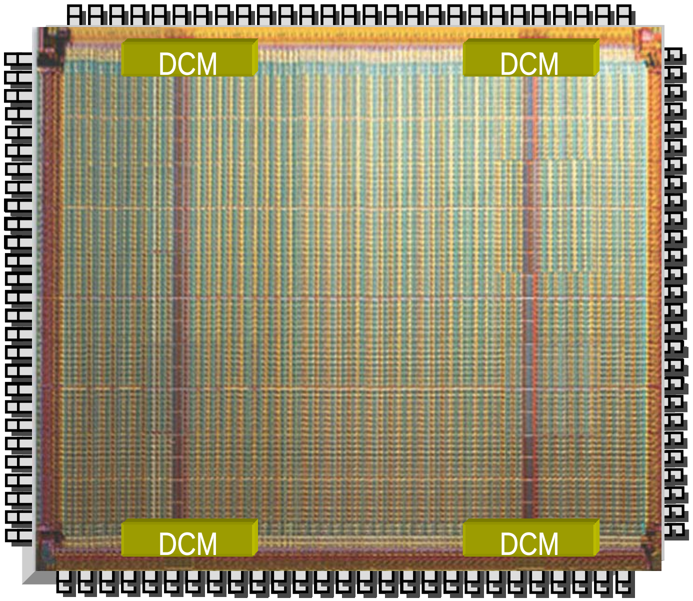
System Interface Summary

- SelectIO supports 24 IEEE/JEDEC standards.
- Flexible I/O block.
  - Programmable slew rate.
  - Independent input, output and programmable 3-state registers.
  - Input delay for 0 hold time.

Xilinx Spartan-3 Family

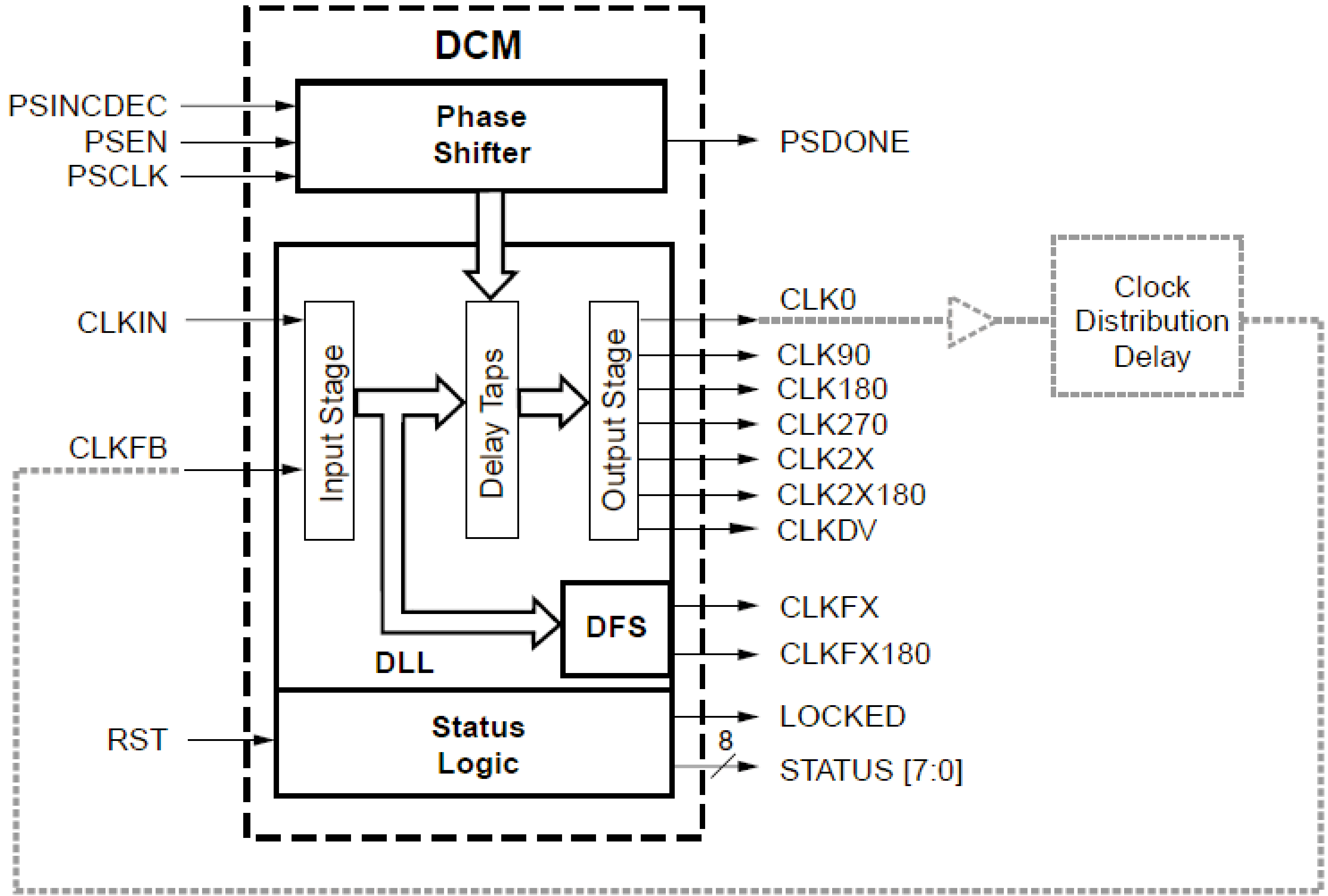
- Programmable Input Output Blocks (IOB).
- Clock Management Blocks (DCM).
- Configurable Logic Blocks (CLB).
- Flexible Synchronous Memory (BlockRAM).
- A variety of programmable routing resources.

Digital Clock Manager (DCM)



- Delay Locked Loop (DLL)
  - Clock phase de-skew.
  - 50% duty cycle correction.
  - 25 MHz to 280 MHz.
  - Simple phase shifts.
- Digital Phase Shift (DPS)
  - (Period / 256) increments.
- Digital Frequency Synthesis (DFS)
  - M/N clock multiply and divide.
  - M= 2 to 32, N= 1 to 32

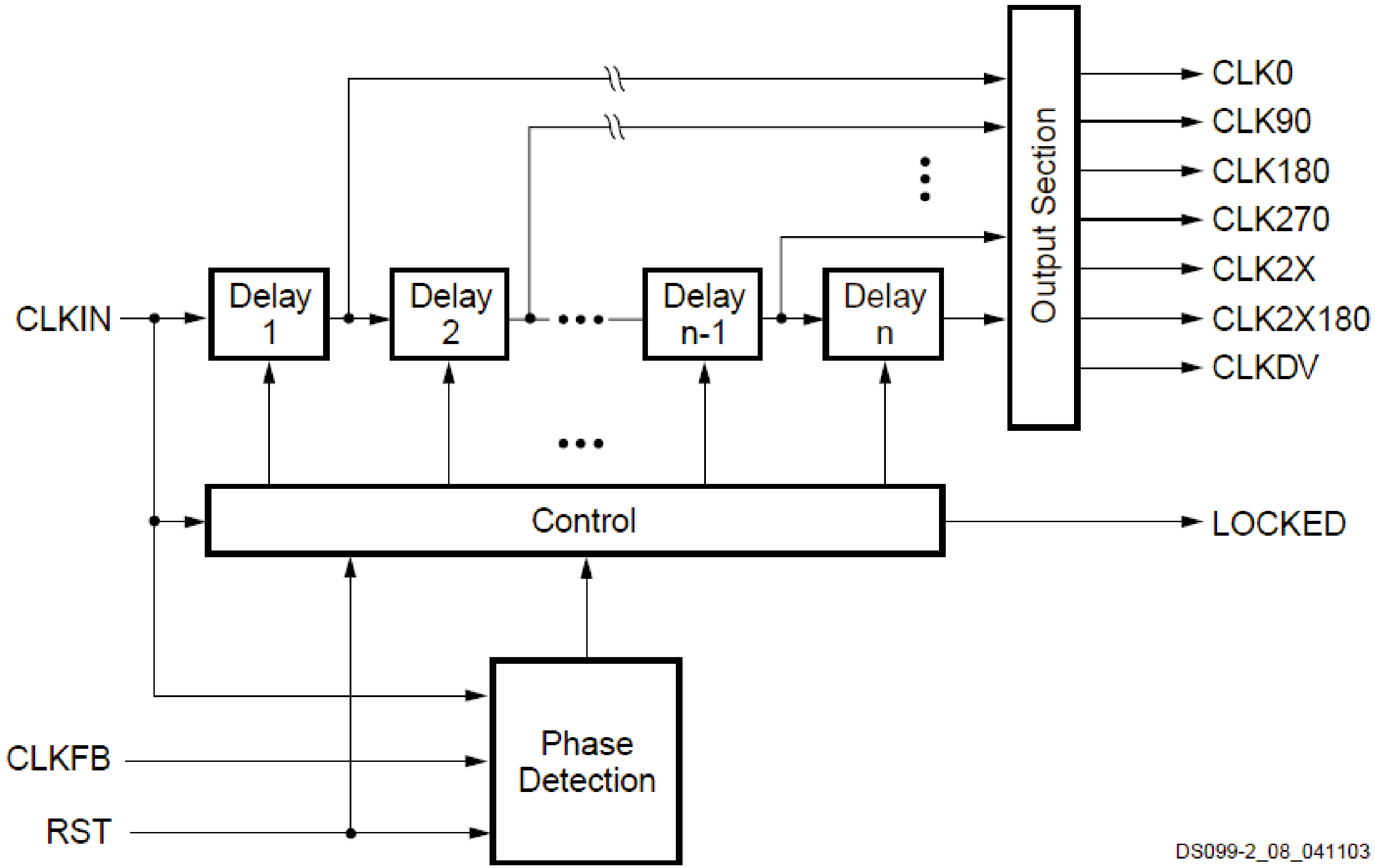
DCM Functional Blocks



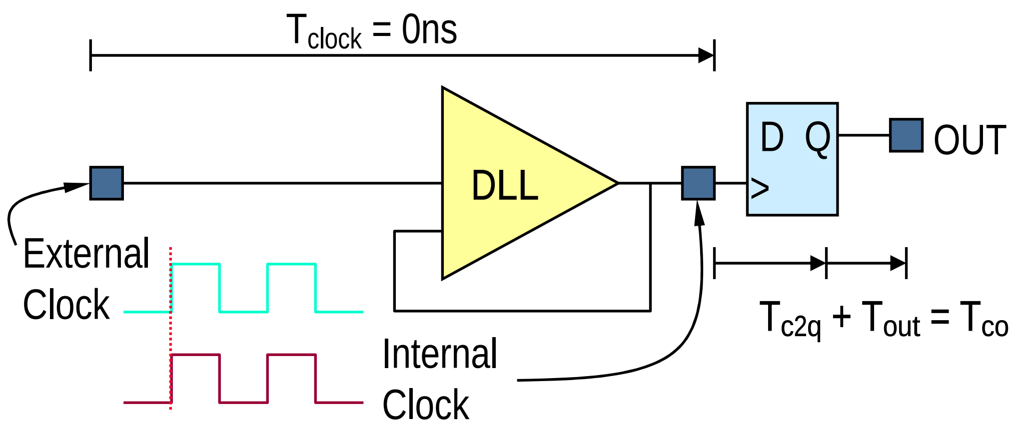
Delay Locked Loop (DLL)

- A DLL inserts delay on the clock net until the clock input rising edge is in phase with the clock feedback rising edge.
- With a well-designed clock distribution network, the clock edges arrive simultaneously everywhere in the part concurrent with their arrival on the clock input pin.

DLL Functional Blocks



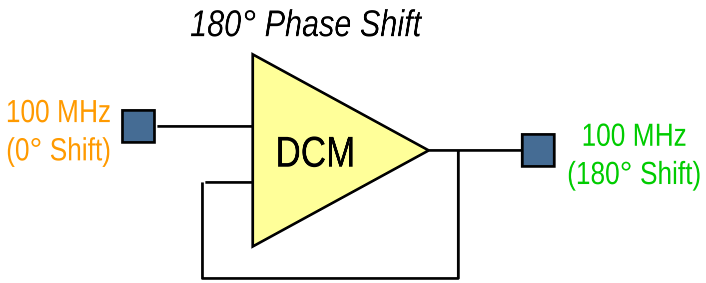
DLL: Adjust I/O Timing

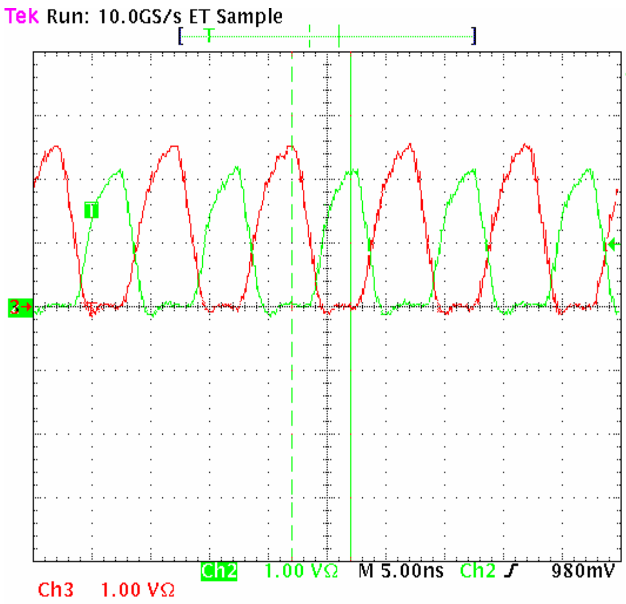


- Eliminate clock distribution delay.
  - External clock pin and internal clock are aligned.
- Optional duty cycle correction.
  - 50/50 duty cycle correction applied when specified.

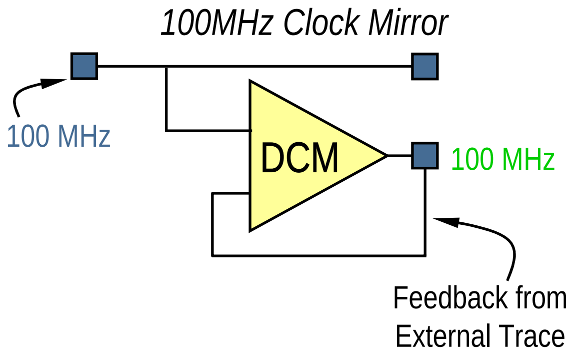
DLL: Phase Shift

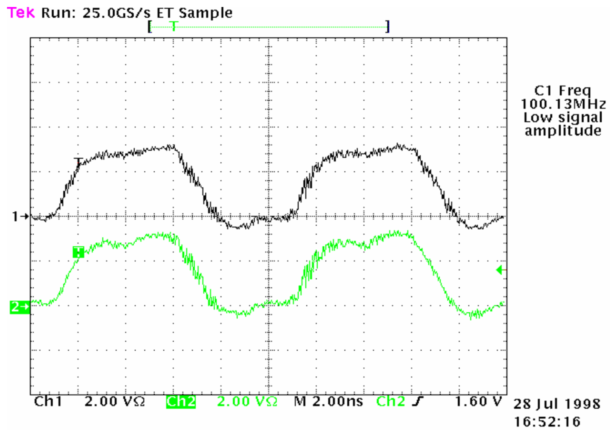
- DLL phase shifts: 0°, 90°, 180°, and 270°.
- Increase performance by utilizing additional clock phases.
- 50/50 duty cycle correction available.
- Excellent for external memory interfaces.

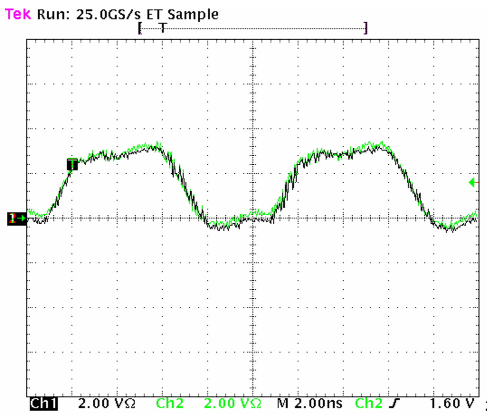




DLL: Clock Mirrors







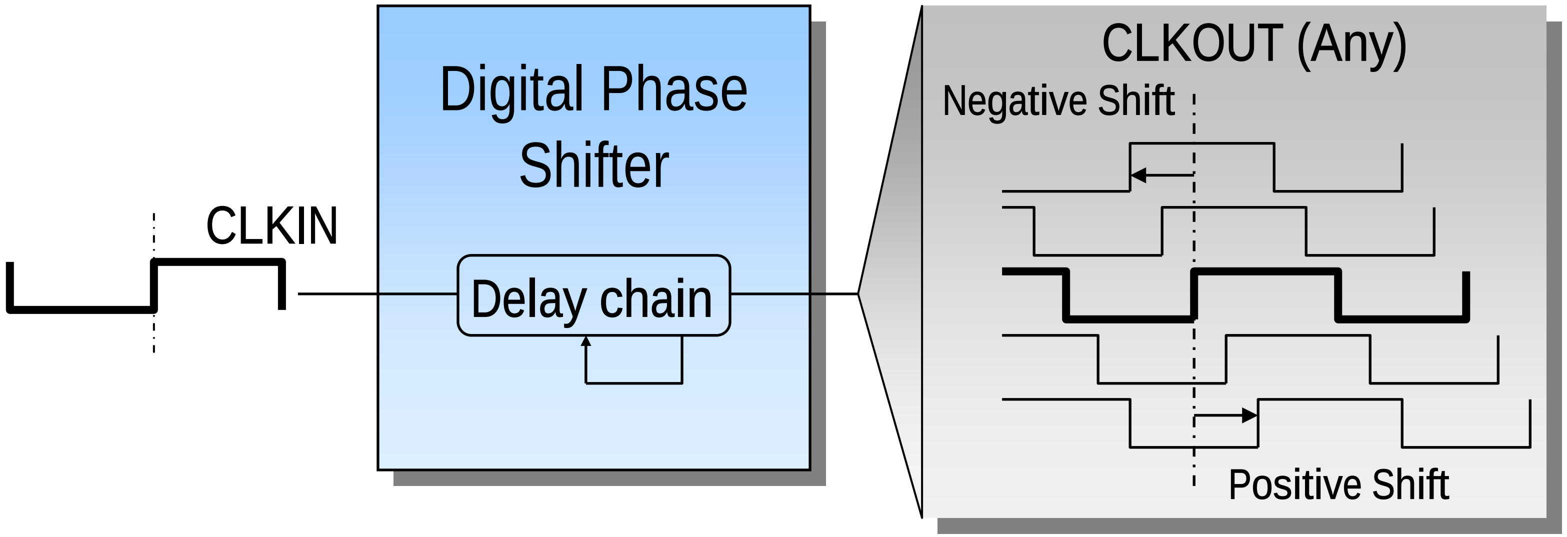
*Actual Device Measurements

- Input clock duplication.
  - Provides on and off-chip clocks.
  - Clock distribution across system.
  - Extremely low output skew.
- Cleans up backplane or noisy clocks.

DLL: Frequency Adjustment

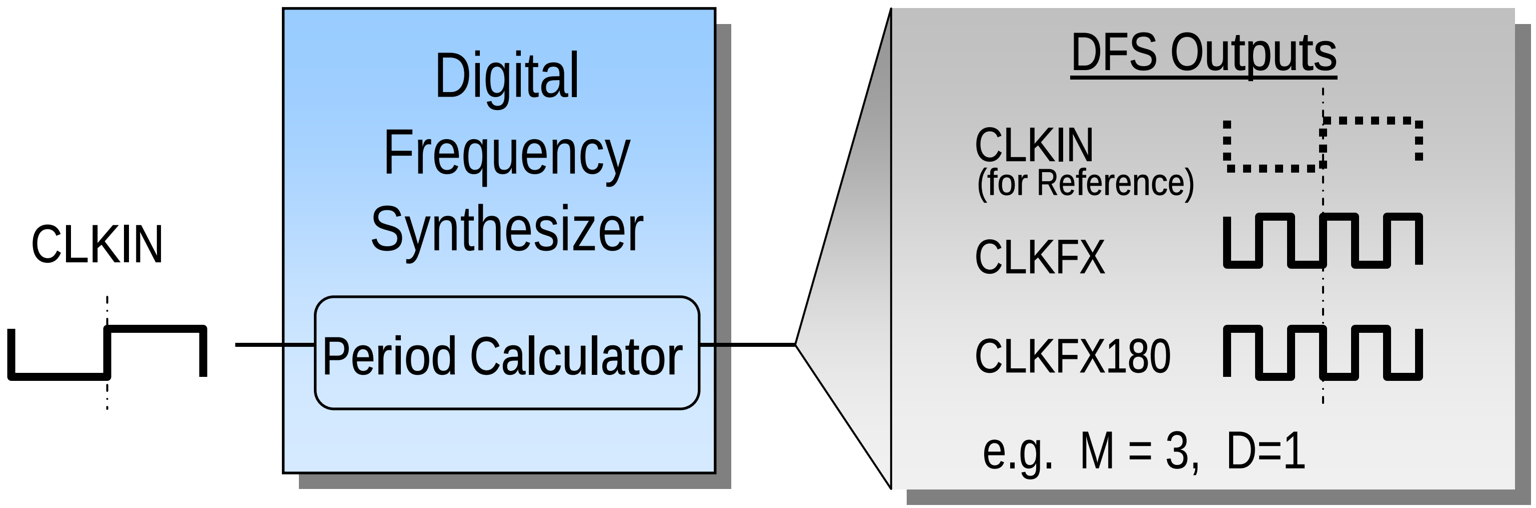
- Frequency multiplication by 2.
- Selectable division values from 1.5 to 16.
- Cascade to combine functions.
- 50/50 duty cycle correction available.

Digital Phase Shifter (DPS)



- Place clock edge anywhere within +/- clock period.
  - Amount of phase shift = (PS/256) x period
  - Where -255 < PS < +255
- Fixed or variable modes.
- Phase shift constant across temperature and voltage.
- Phase shift affects all DCM outputs.

Digital Frequency Synthesizer (DFS)



- Synthesize any frequency within DFS operating range.
  - CLKOUT = (M ÷ D) x CLKIN
  - M = 2 to 32 and D = 1 to 32
- Output frequency constant across temperature and voltage.
- Outputs have 50/50 duty cycle.

DCM Clock Options

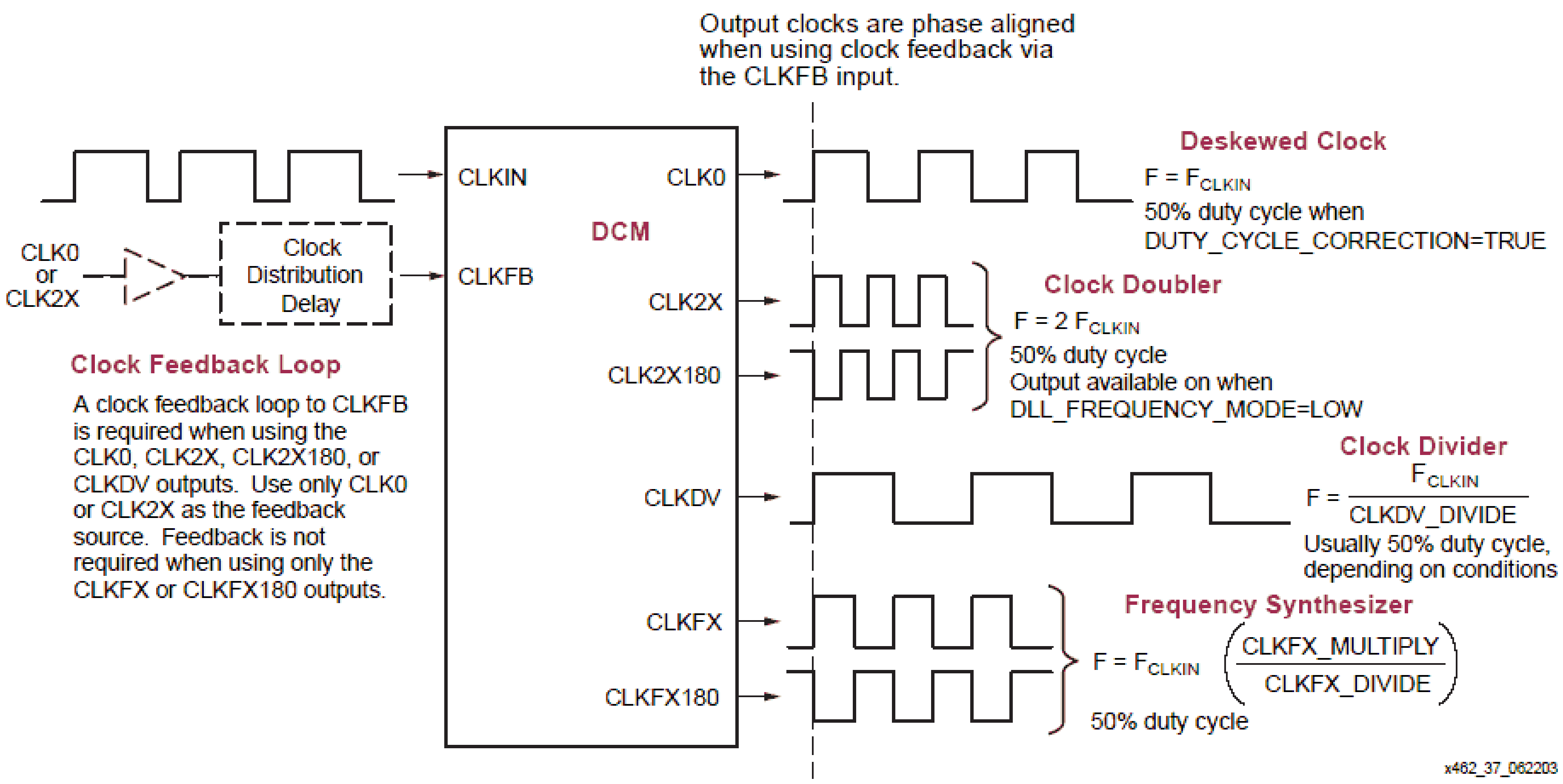


Figure 37: **Clock Synthesis Options**

Clock Management Summary

- All digital DLL implementation.
  - Clock deskew.
  - Input noise rejection.
  - 50/50 duty cycle correction.
  - Clock mirroring.
- Multiply or divide clock.
- Programmable phase shift.

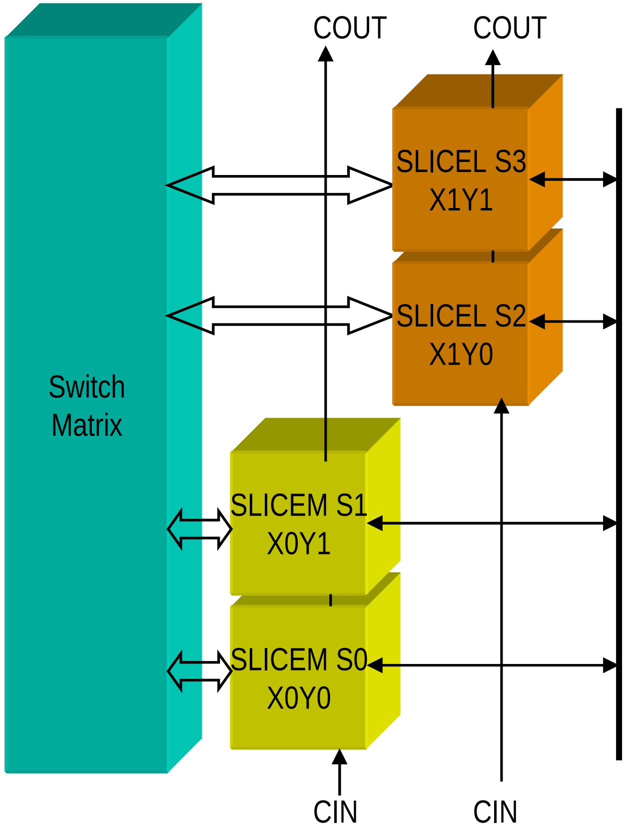
Lecture #7 Agenda

- Spartan-3 FPGA architecture detail part two.
- Technical information and diagrams reproduced with permission from Xilinx.

Xilinx Spartan-3 Family

- Programmable Input Output Blocks (IOB).
- Clock Management Blocks (DCM).
- Configurable Logic Blocks (CLB).
- Flexible Synchronous Memory (BlockRAM).
- A variety of programmable routing resources.

Configurable Logic Block (CLB)



- Switch matrix connects to routing.
- Four slices per CLB.
  - 2 SLICEL are Logic only.
  - 2 SLICEM are Logic / Memory.
- Fast arithmetic functions with cascadable look-ahead carry chains.

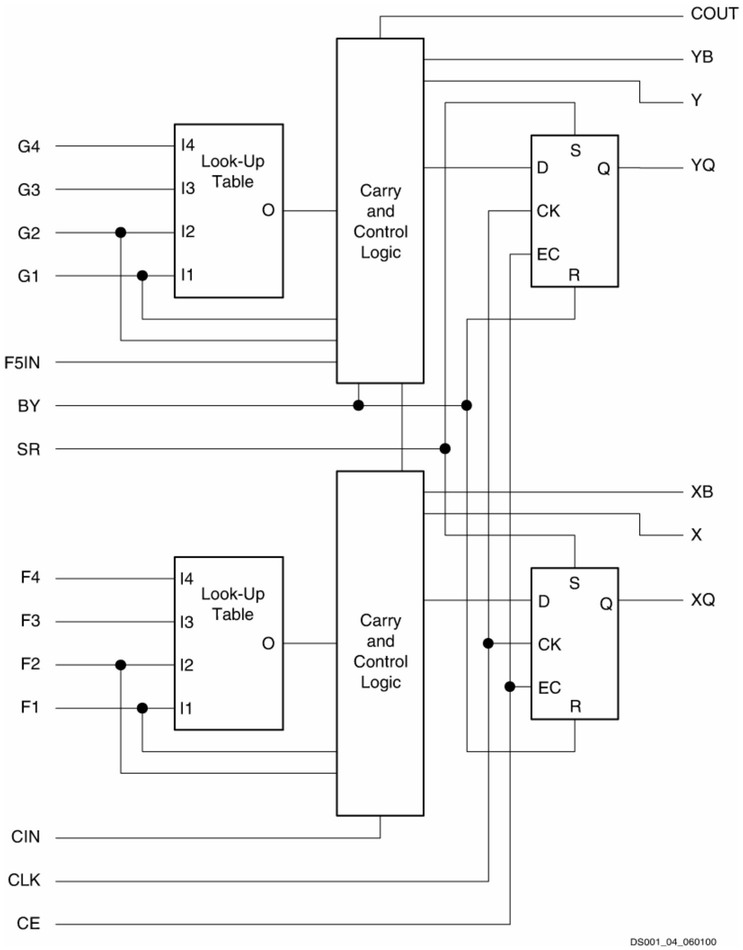
Spartan-3 Slice Capabilities

- Basic SLICEL structure of a slice is two 4-input look-up tables followed by two D flip-flops (plus extra stuff).
- Basic SLICEM structure is like SLICEL but the LUT4s may instead be used as RAM or a shift register.

SLICEM	Function	SLICEL
✓	Logic/ROM	✓
✓	Arithmetic/Carry	✓
✓	Wide Mux	✓
✓	Distributed RAM	
✓	Shift Register	

Spartan-3 Slice Capabilities

- Four-input LUT
  - Any 4-input logic function
  - 16-bit x 1 RAM (SLICEM)
  - 16-bit shift register (SLICEM)
- Carry & Control
  - Fast arithmetic logic
  - Multiplier logic
  - Multiplexer logic
- Storage element
  - Latch or flip-flop
  - Set and reset
  - True or inverted inputs
  - Sync. or async. control

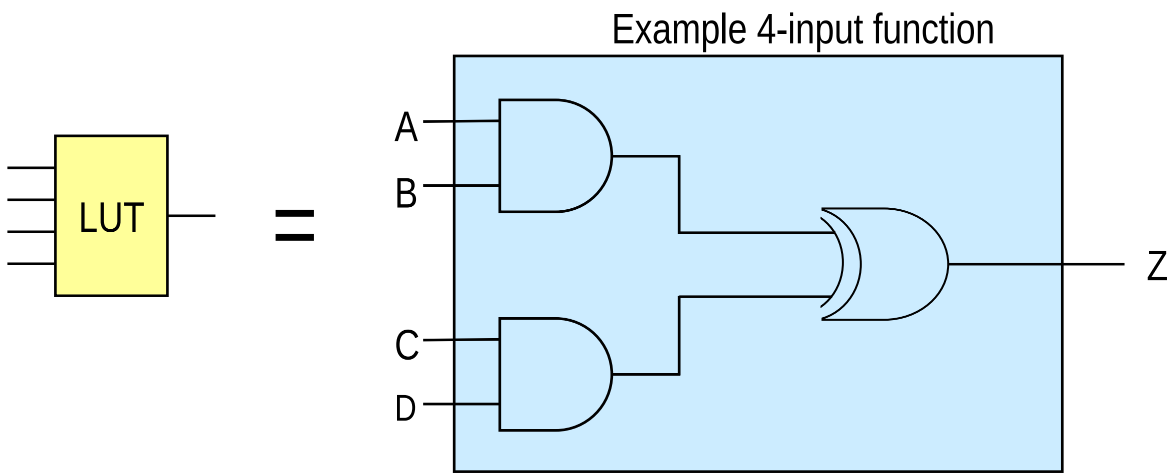


Four-Input LUT

- Implements combinational logic.
  - Any function of 4 or fewer inputs.
  - Cascaded for wide-input functions.

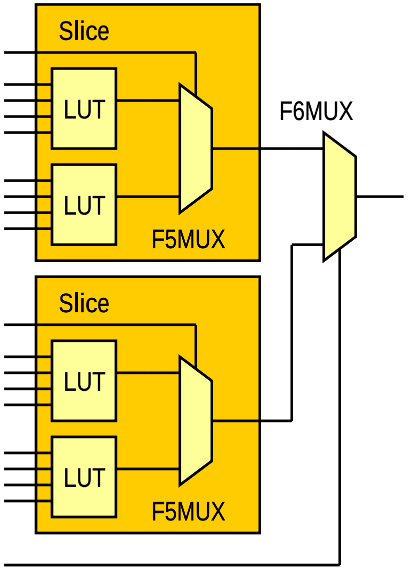
Truth Table

Inputs(ABCD)	Output(Z)
0000	0
0001	0
0010	1
0011	0
……	..
1110	1
1111	1



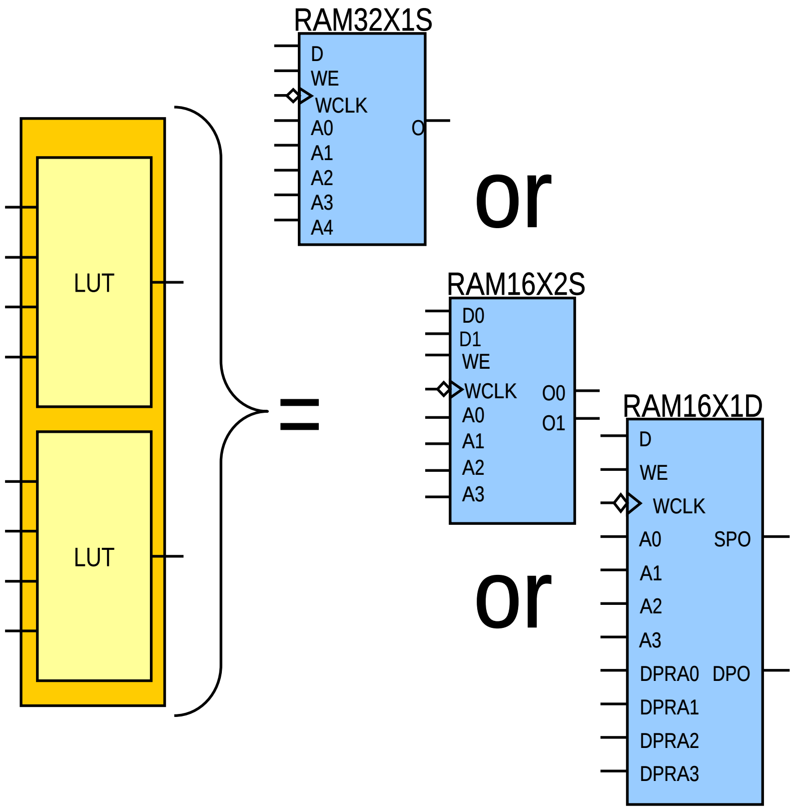
Dedicated Multiplexers

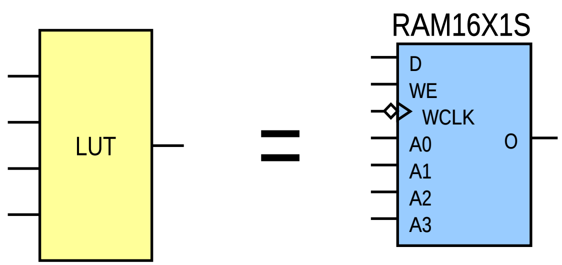
- More efficient than multiplexers implemented with look-up tables.
  - F5MUX used with LUT outputs.
  - F6MUX used with SLICE outputs.
  - F7MUX used with CLB outputs.
  - F8MUX used with F7MUX outputs.
- Efficient way to build wide muxes and functions up to eight inputs.



Distributed RAM

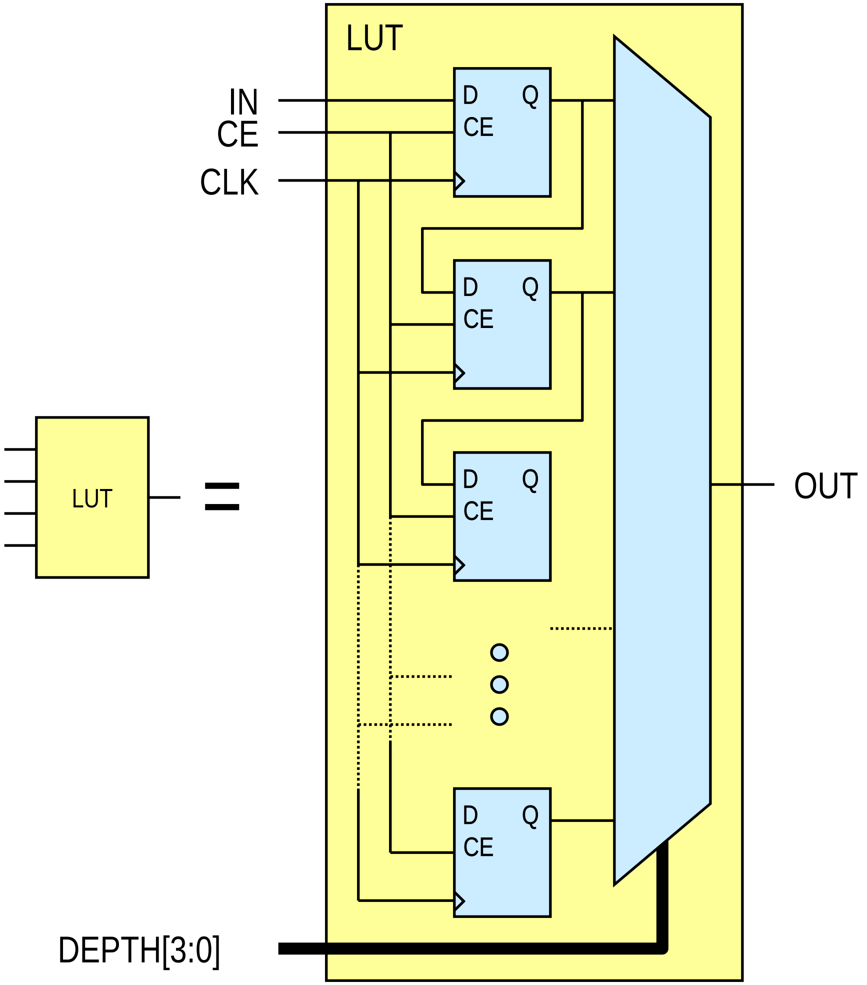
- A LUT in a SLICEM may be configured for use as a RAM.
  - Implement single and dual port.
  - Cascade LUTs to increase size.
- Synchronous write only.
- Reads may be synchronous or asynchronous.





Shift Register

- A LUT in a SLICEM may be configured for use as a RAM.
  - Implement single and dual port.
  - Cascade LUTs to increase size.
  - Dynamically addressable delay up to 16 cycles.

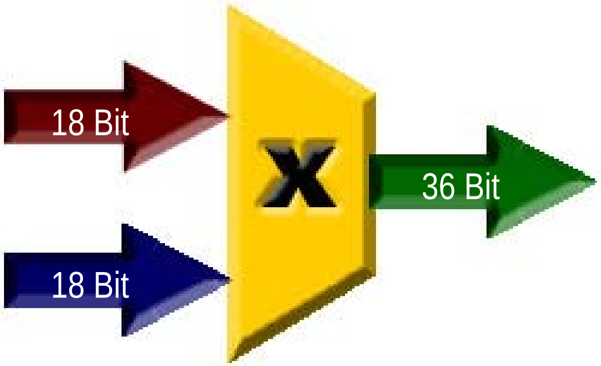


Arithmetic / Carry Logic

- Dedicated look-ahead carry logic.
  - High performance for counters and arithmetic functions.
  - Can be used to cascade LUTs for wide-input logic functions.
- Resources for efficient LUT implementation of shift and add multipliers.

Embedded Multipliers

- Not actually located in CLB, but this seems a good place to bring it up…
  - 18 x 18 bit signed operation.
  - 17 x 17 bit unsigned operation.
  - 2's complement operation.
  - Combinational and pipelined options.



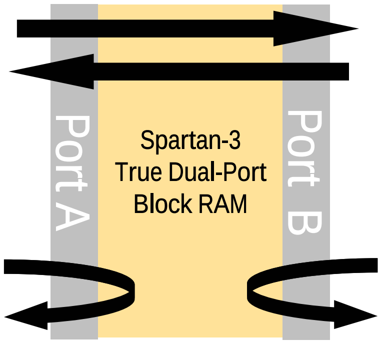
Spartan-3 CLB Summary

- Flexible Configurable Logic Block (CLB).
  - Logic, Flip Flops.
  - Distributed RAM, Shift Registers.
- CLB configurable for simple to complex logic.
  - Any 6 input function into one logic level.
- Excellent for fast arithmetic operations.
  - Specialized carry logic for arithmetic operations.
  - Fast DSP functions, FIR filters.

Xilinx Spartan-3 Family

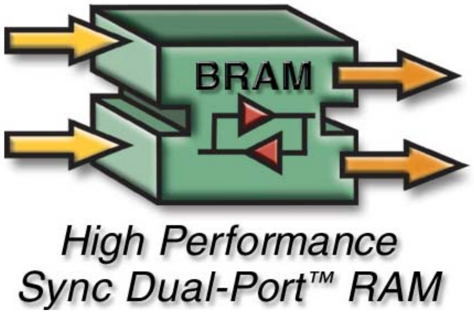
- Programmable Input Output Blocks (IOB).
- Clock Management Blocks (DCM).
- Configurable Logic Blocks (CLB).
- Flexible Synchronous Memory (BlockRAM).
- A variety of programmable routing resources.

BlockRAM



- Dedicated blocks of 18-kilobit synchronous RAM.
- Ideal for many memory requirements.
- Builds both single and true dual-port memories, true dual port ideal for asynchronous FIFOs.
- May be initialized and used as synchronous ROM.

BlockRAM

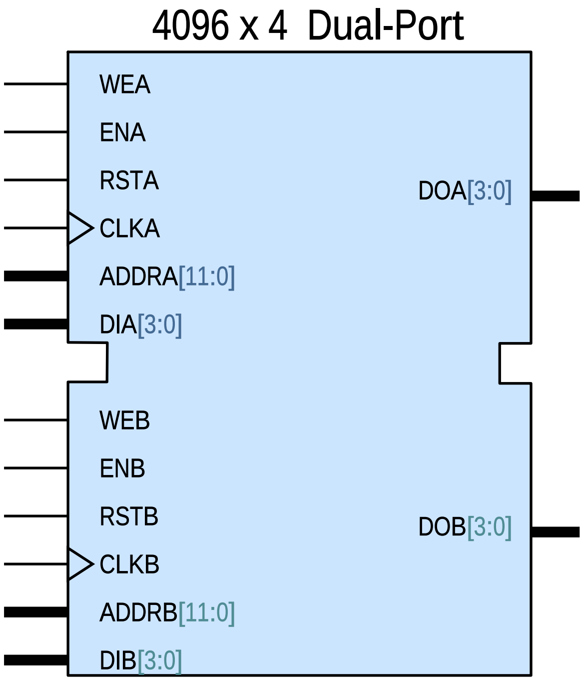


- Independent configuration for port A and for port B.
- Enables data width conversion.

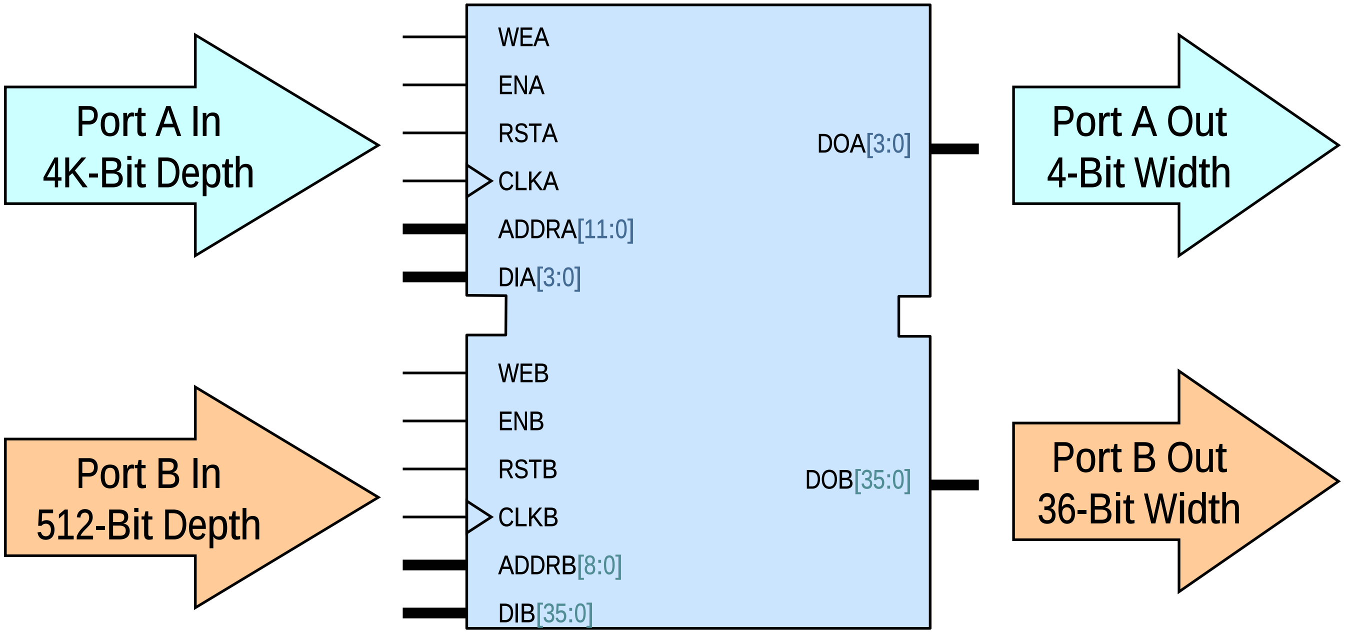
Configuration	Depth	Data bits	Parity bits
16K x 1	16Kb	1	0
8K x 2	8Kb	2	0
4K x 4	4Kb	4	0
2K x 9	2Kb	8	1
1K x 18	1Kb	16	2
512 x 36	512	32	4

True Dual-Port

- True simultaneous read and/or write to both ports.
- Each port has independent controls.
  - Address
  - Clock/Enable
  - Data
  - Read/Write
  - Reset
- May be used as two independent half-sized single port memories.



Dual-Port Flexibility



- Each port can be configured with different data width.
- Provides easy data width conversion.

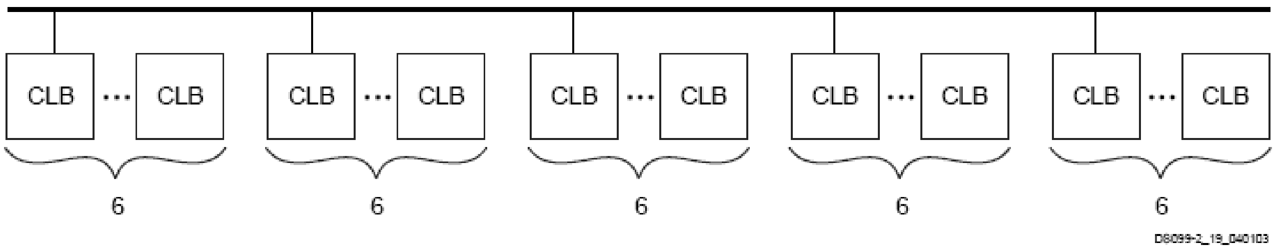
Embedded Memory Summary

- Flexible BlockRAMs enable:
  - Single and True Dual-Port RAMs.
  - FIFOs for buffering data.
  - Data width conversion.
  - Caches and register banks.

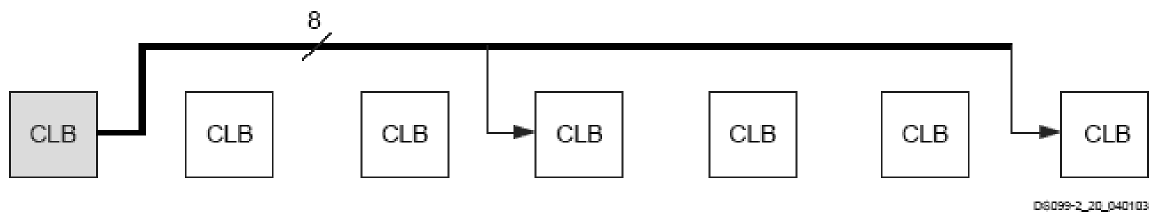
Xilinx Spartan-3 Family

- Programmable Input Output Blocks (IOB).
- Clock Management Blocks (DCM).
- Configurable Logic Blocks (CLB).
- Flexible Synchronous Memory (BlockRAM).
- A variety of programmable routing resources.

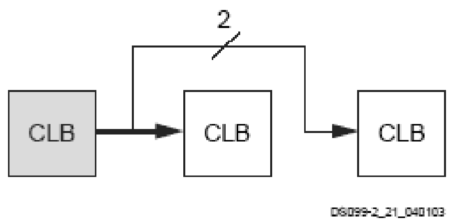
Routing Wire Types



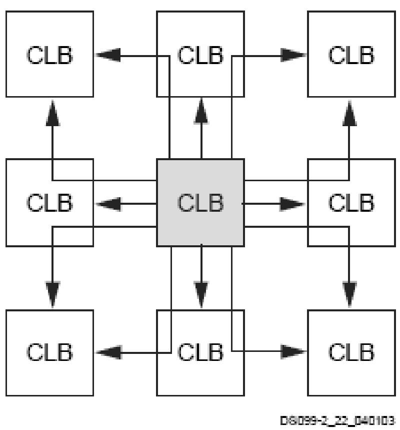
(a) Long Line



(b) Hex Line



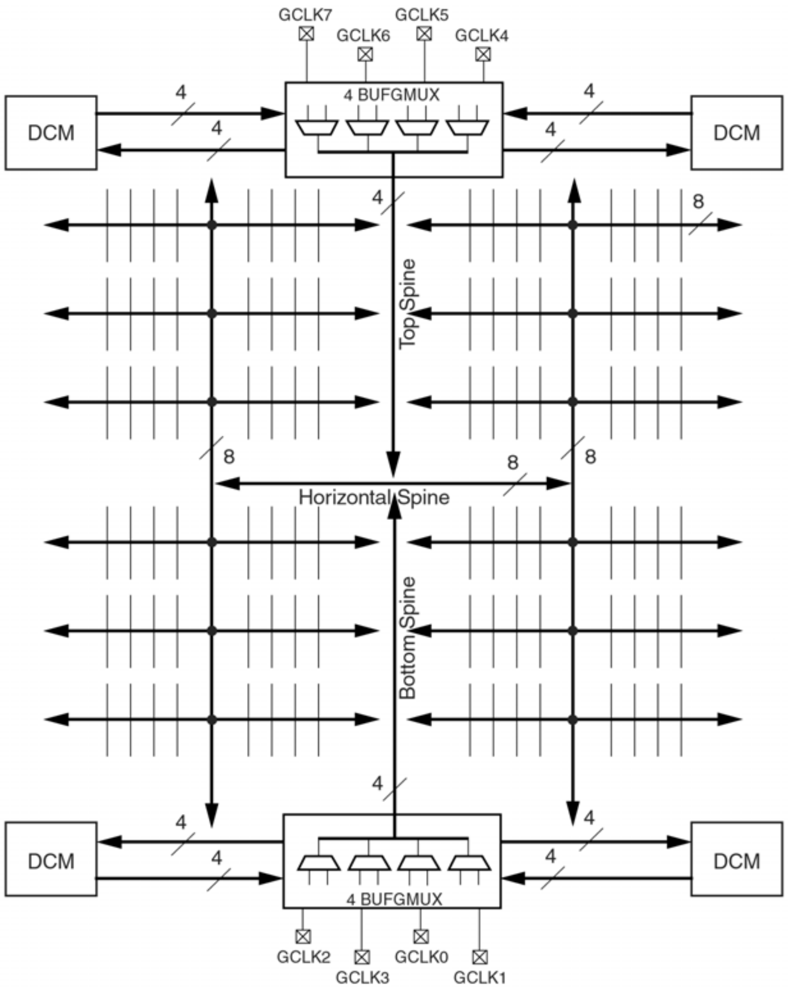
(c) Double Line



(d) Direct Lines

Global Routing

- Distribute clocks and other high fanout signals throughout the device with minimum skew.
- Eight global clock nets designed to distribute high fanout clock signals.



Routing Summary

- Vector-based routing provides predictable routing delays independent of:
  - Design placement.
  - Device size.
- Superior routing results in quick routing times and increased design performance.