Department of Electrical
and Computer Engineering

INTRODUCTION TO CALD
(Computer Aided Logic Design)

Introduction

- Late 1960's-early 80's – most logic design carried out using the TTL (transistor transistor logic) or the CMOS (complementary metal oxide transistor) logic families. This design philosophy is now know as discrete logic design (although at the time of use it would not have been).

- 1980's – growth of the PLA (Programmable Logic Array) and PAL (Programmable Array Logic) chip technology. Allowed a number of product terms to be once only programmed into a single integrated circuit. Only implemented combinational circuits. See Figure 1 and Figure 2.

- The next generation of the PLA/PAL technology allowed the circuits to be programmed in the field and then erased and reprogrammed a number of times. Usually based on EPROM type technology for erasure. Flip flops also started to appear in on the outputs of some devices allowing sequential circuits to be built. See Figure 3 and Figure 4.

- Late 1980's – more sophisticated devices started to appear. These were often known as EPLDs (Electrically Programmable Logic Devices) based on macrocell architectures, or FPGAs (Field Programmable Gate Arrays) based on interconnected gates, or low level structures. These devices contained many more flip flops and more combinational logic capability compared to the earlier PLA/PAL type of circuitry.

- 1990's – Devices now allow almost arbitrary logic to be implemented. Very large devices, equivalent to 150,000 logic gates now available and the size of the devices is increasing.

Department of Electrical
and Computer Engineering

PLA logic

- Used for combinational circuits

Function implemented below:

F1=AB'+AC

F2=AC+BC

.

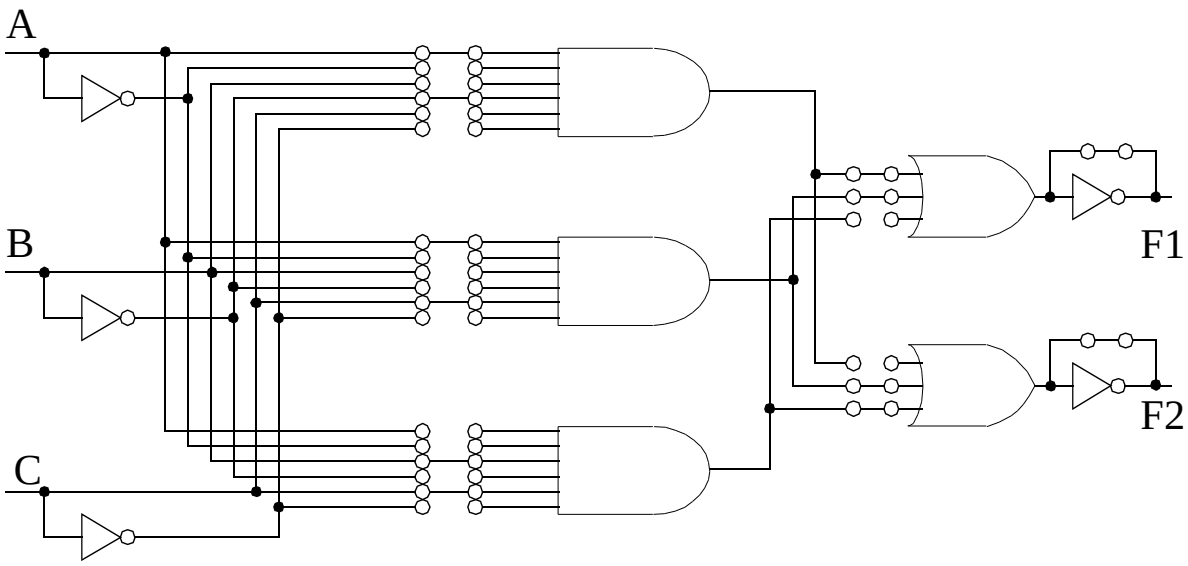


Figure 1 : Typical PLA layout

- Note that the PLA shown above is not typical – it is much too small to be a practical PLA. More typical PLA may have 50 product terms, 10 inputs and 8 outputs.

- One must attempt to minimise the number of product terms so that the limited number of AND gates are used.

- Has programmable AND and OR arrays in the device.

Department of Electrical
and Computer Engineering

PAL logic

- Used to implement combinational circuits
- Only has a programmable AND array – OR array is fixed. Easier to program than the PLA but not a flexible.

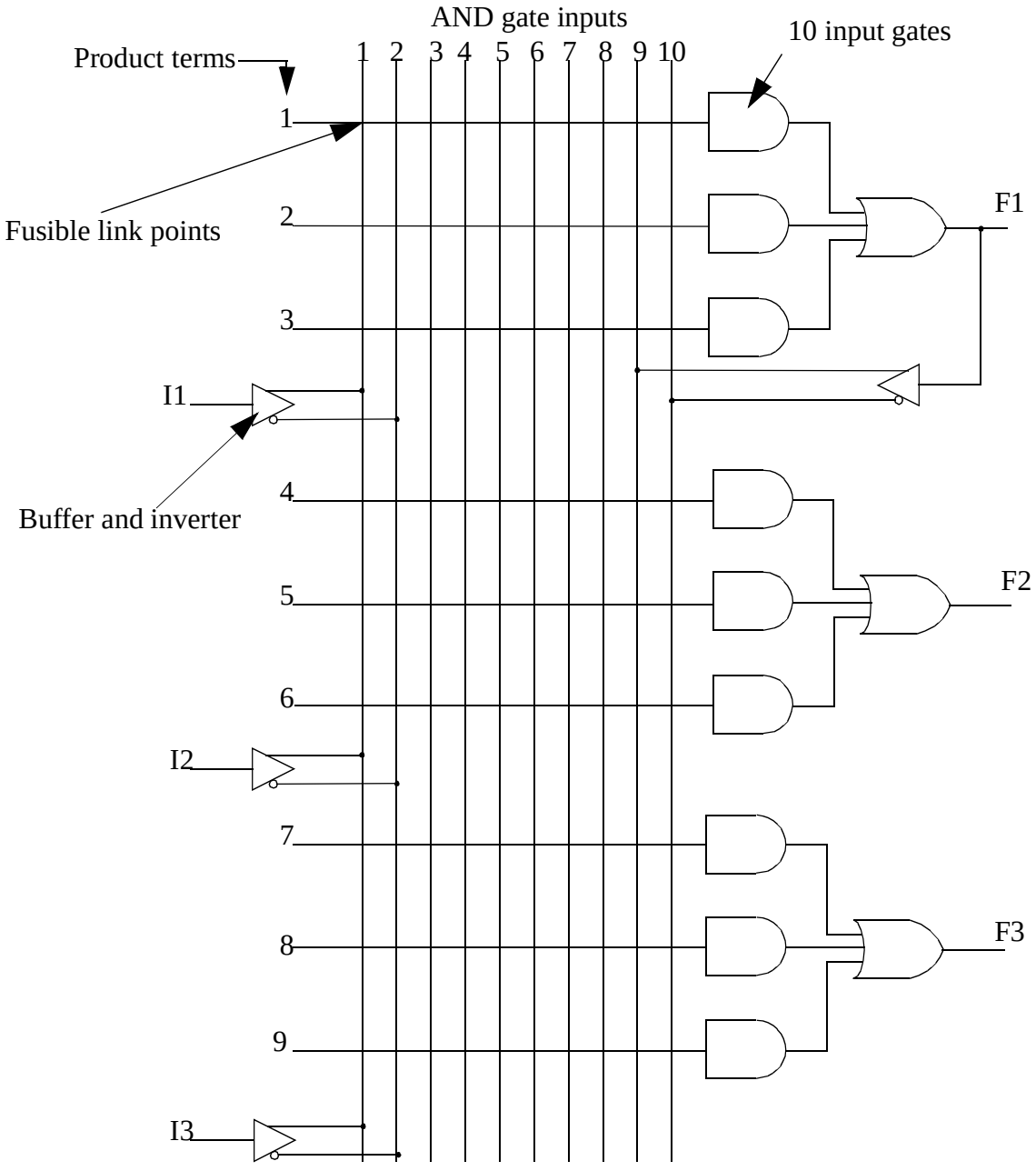


Figure 2 : Classic PAL logic structure

Department of Electrical
and Computer Engineering

Larger PLA/PAL logic families

- Now contain sequential elements.
- Example: Altera Classic EPLD series of devices – based on a macrocell architecture

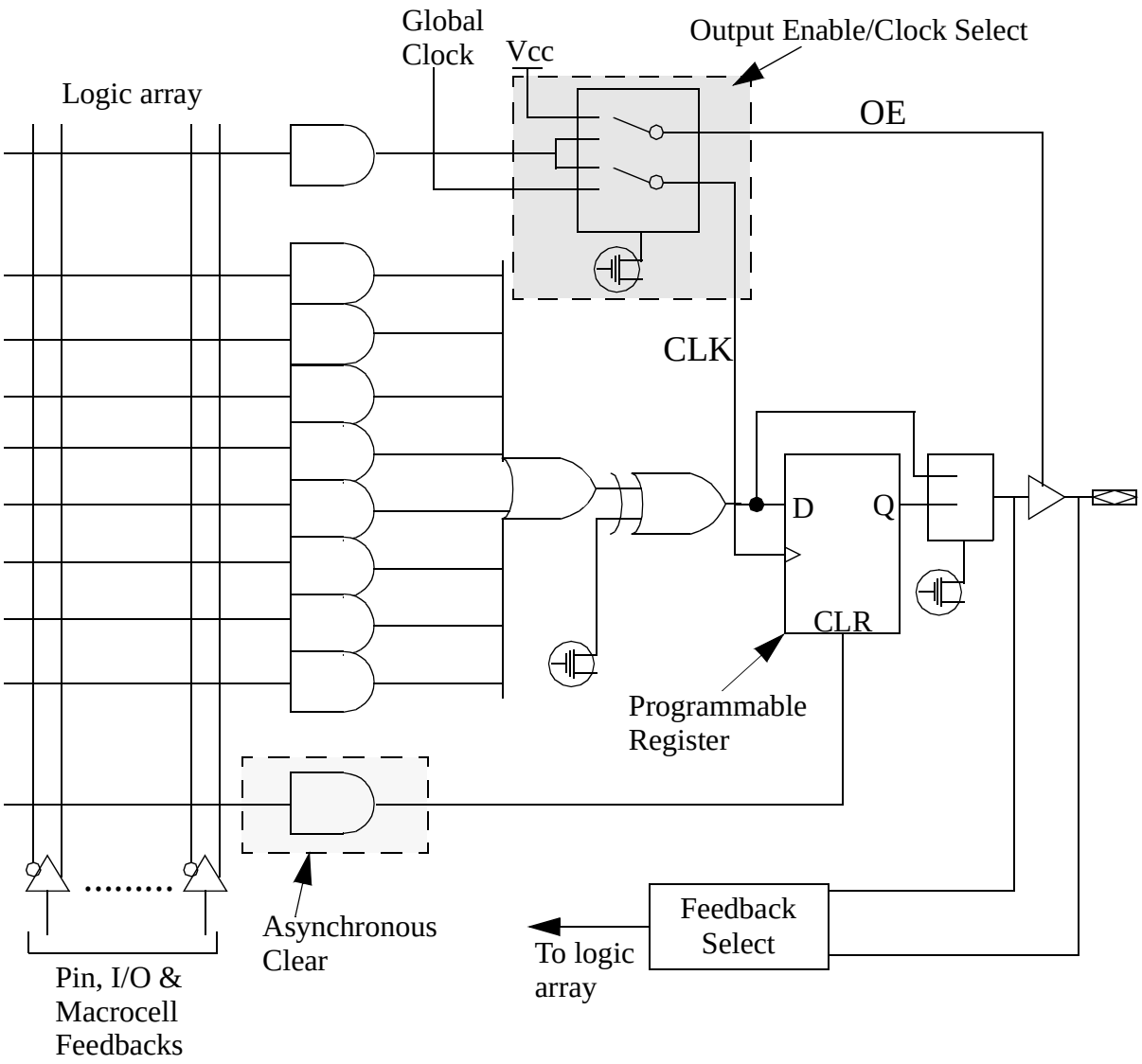


Figure 3 : Altera Classic Macrocell

Some notes

- Individual cells can be programmed for either combinational logic or sequential logic.
- Based on a PAL AND/OR array i.e. the AND array is programmable.
- If an I/O pin is used as an input then it cannot be used as a macrocell output.
- Macrocell register can be programmed to be a D, T, JK, or SR flip flop.
- Dedicated inputs are globally available to all macrocells.
- Feedback multiplexer may make the feedback signals available or in the quadrant of the macrocell (depends on the particular part).

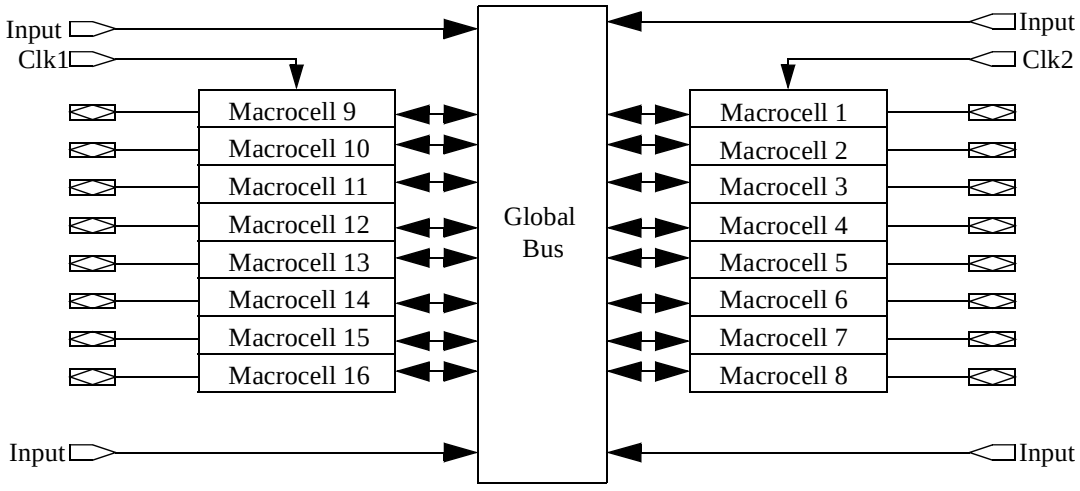


Figure 4 : Internal block diagram of the Altera EP610 EPLD.

Department of Electrical
and Computer Engineering

Enhanced Larger PLA/PAL Logic Family Devices

- These EPLDs are essentially an extension of the previous logic family.
- Based on using logic array blocks (LABs) – a sort of super macro-cell.
- Have more flexible interconnect structures.

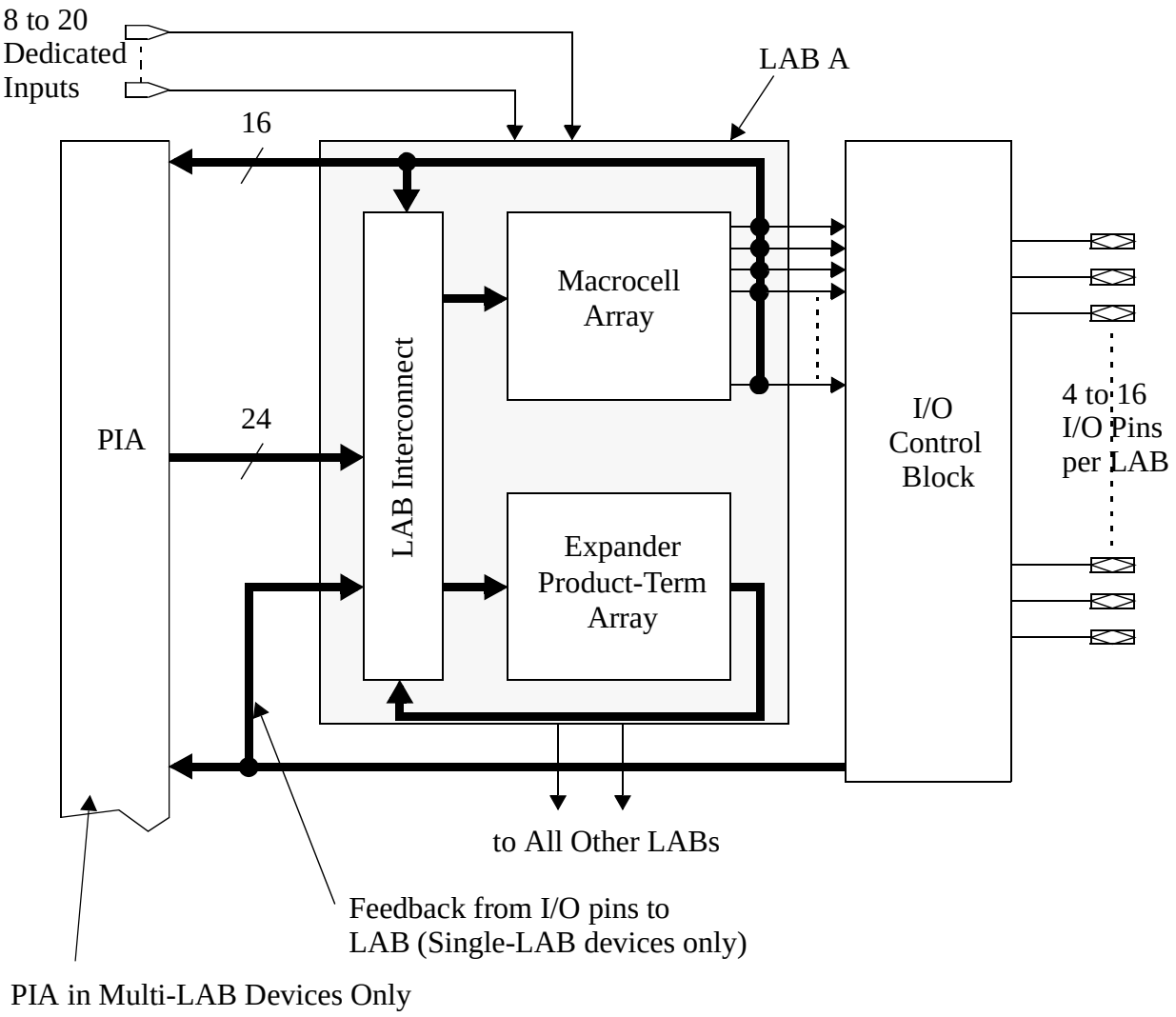


Figure 5 : Altera MAX 5000 Architecture

- The expanders are to complement the capabilities of the macrocell. The expander product-term array consists of a group of unallocated, inverter product terms that can be used and shared by all the macrocells in the LAB to create combinational and registered logic.

- All macrocell outputs are globally routed within a LAB using the LAB interconnect.

- I/O control block consists of programmable tristate buffers and I/O pins.

- The Programmable Interconnect Array (PIA) allows multiple LABs to be connected together.

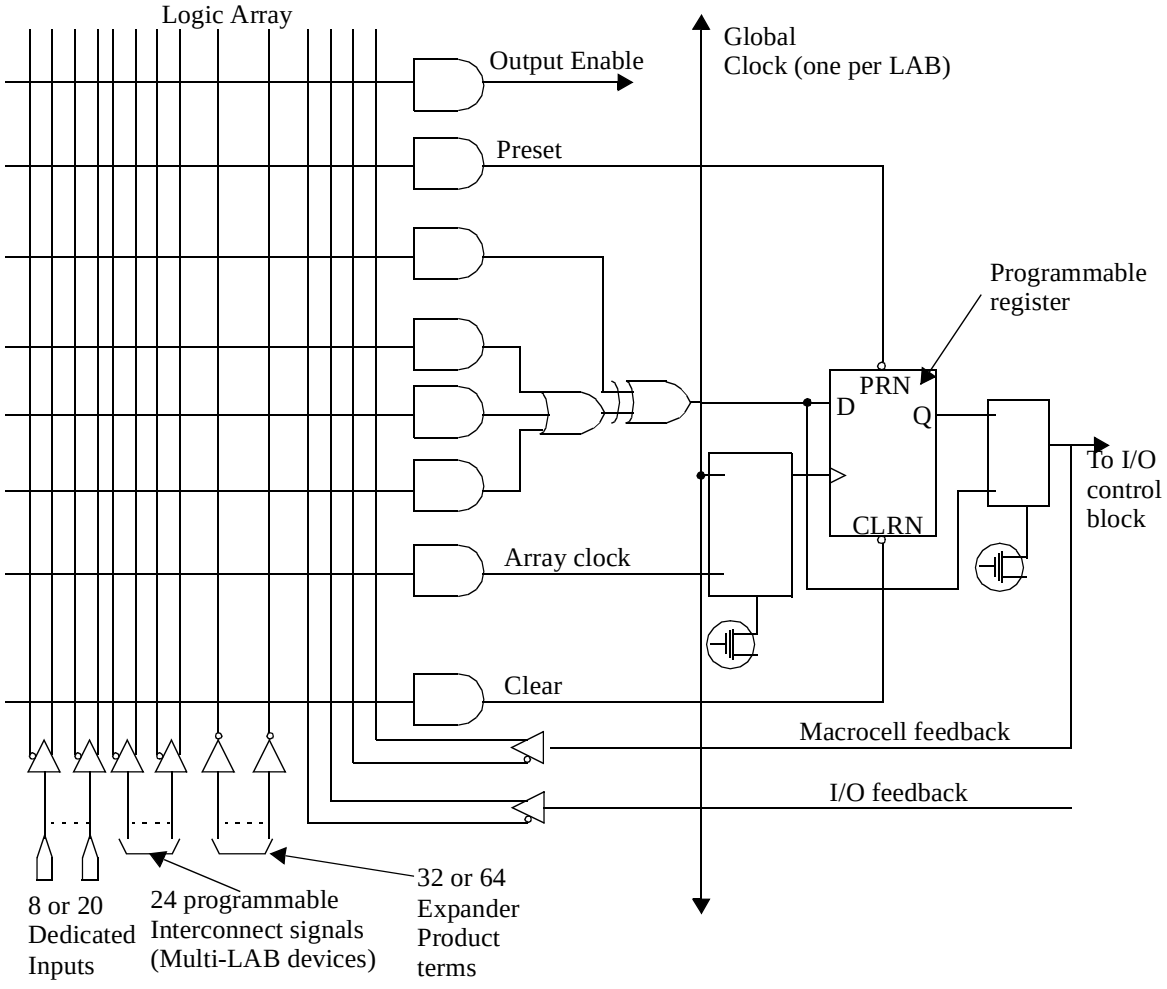


Figure 6 : Altera MAX 5000 Macrocell

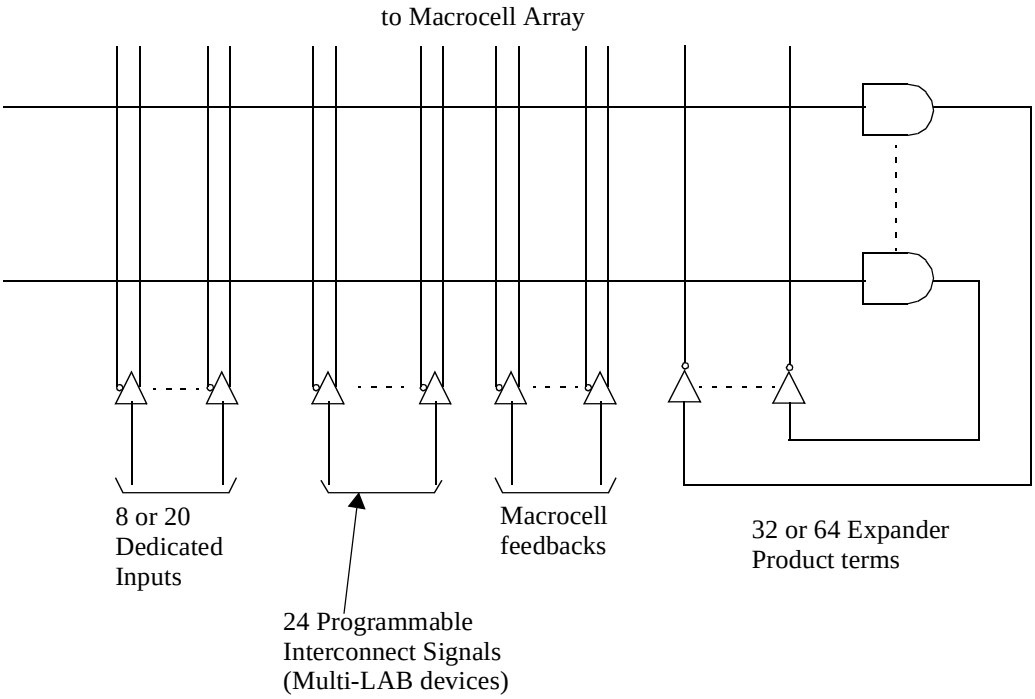


Figure 7 : Expander product terms

Department of Electrical
and Computer Engineering

Modern Devices – Flex 8000

- Shall concentrate on the Flex 8000 series of devices as these will be the devices used in the design project.

- The Flex 8000 (Flexible Logic Element Matrix) series differs from the earlier Altera series of parts as it used SRAM based look-up tables (LUTs) to implement the logic functions, as opposed to the AND/OR structure inherited from PALs, as used in the earlier devices.

- Flex 8000 has up to 50,000 useable gates, 4,752 registers, and 360 I/O pins, depending on the variant of the chip chosen. Device to be used in the project is the EPF8820A which has 8,000 useable gates and 820 flip flops, 84 LABs, 672 logic elements and 152 I/O pins.

- FLEX devices combine the benefits of erasable programmable logic devices (EPLDs) (i.e. high speed and predictable interconnect delays) and field programmable gate arrays (FPGAs) (i.e. fined grained structure and high register count).

Functional Description

- Large matrix of compact building blocks called logic elements (LEs).

- Each LE consists of a 4 input LUT that provides the combinational logic capability, and a programmable register to provide the sequential logic capability (similar to the macrocell block of the earlier families).

- LEs are grouped into sets of eight to create Logic Array Blocks (LABs). Each LAB is an independent structure with common inputs, interconnections and control signals.

- I/O supported by IOEs (Input Output Elements) – located at the end of interconnection rows and columns. Contain bidirectional buffer and a flip flop that can be used as either an input or output register.

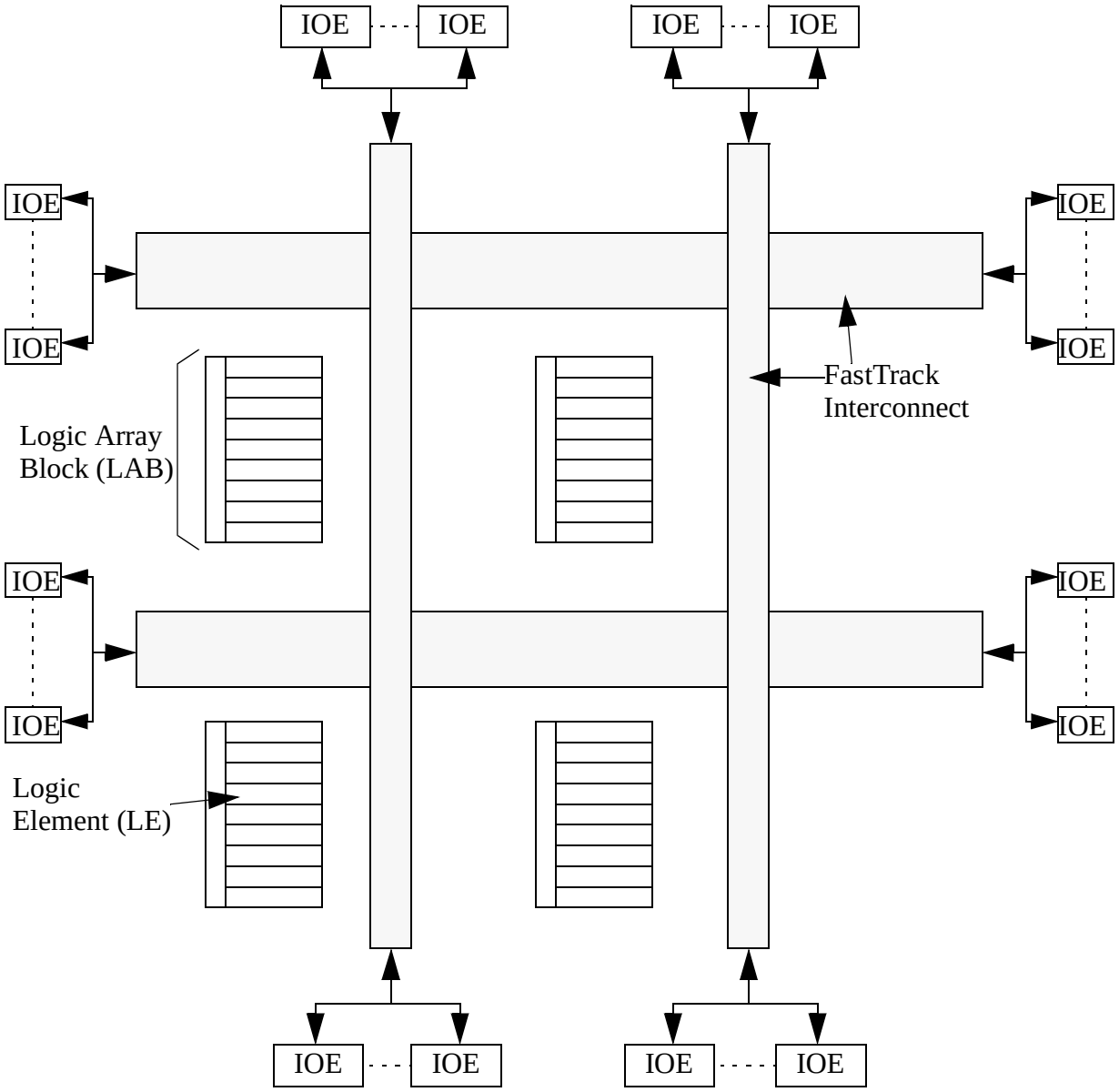


Figure 8 : Flex 8000 Device Block Diagram

Logic Elements

- Smallest element of logic in the Flex 8000. See Figure 9.

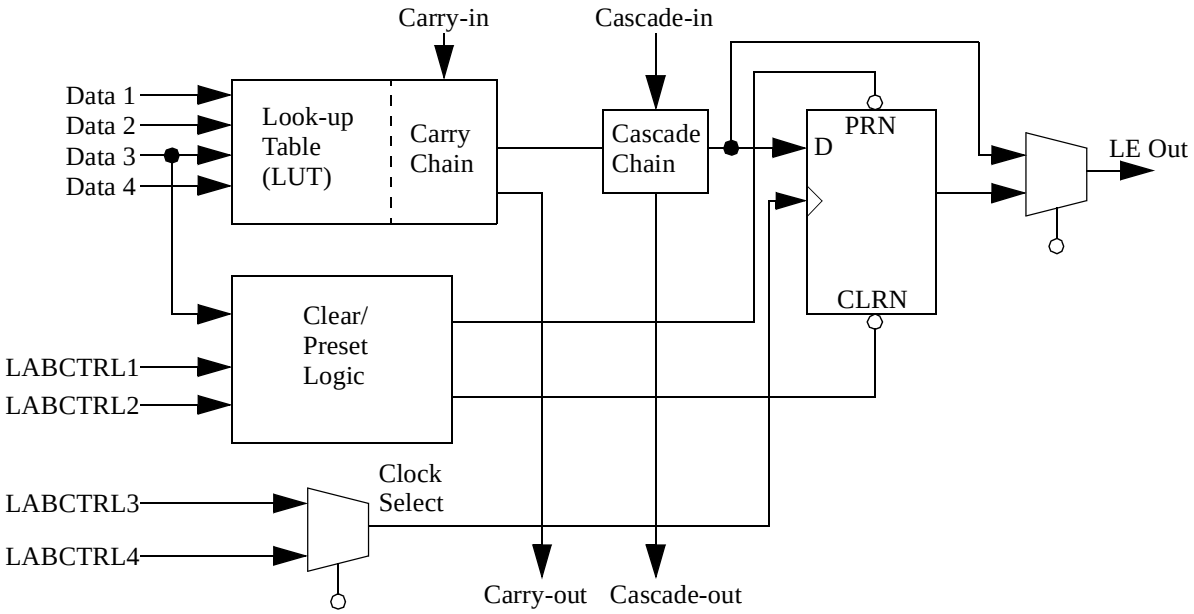


Figure 9 : Flex 8000 Logic Element (LE)

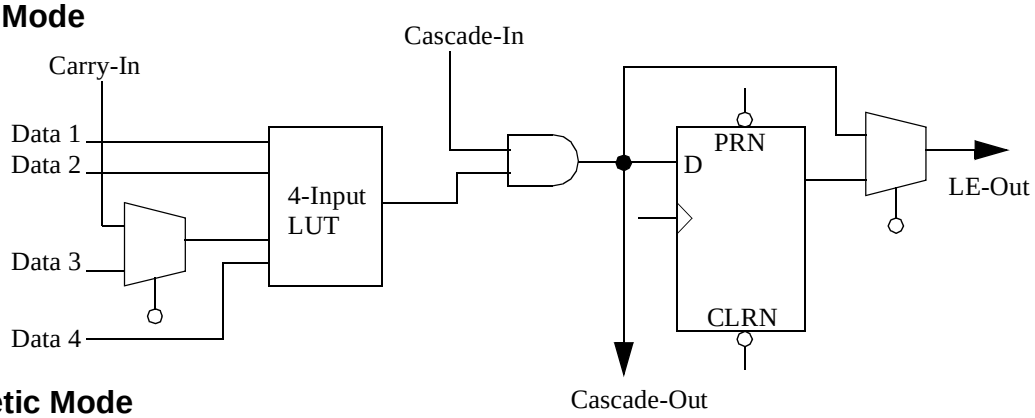
- Two dedicate high speed paths – carry and cascade chains, do not use the general purpose interconnect. Carry for high speed counters and adders, cascade for wide input functions.

- Heavy use of the carry and cascade chains can restrict placement and routing of other logic - should only be used for speed critical portions of the design.

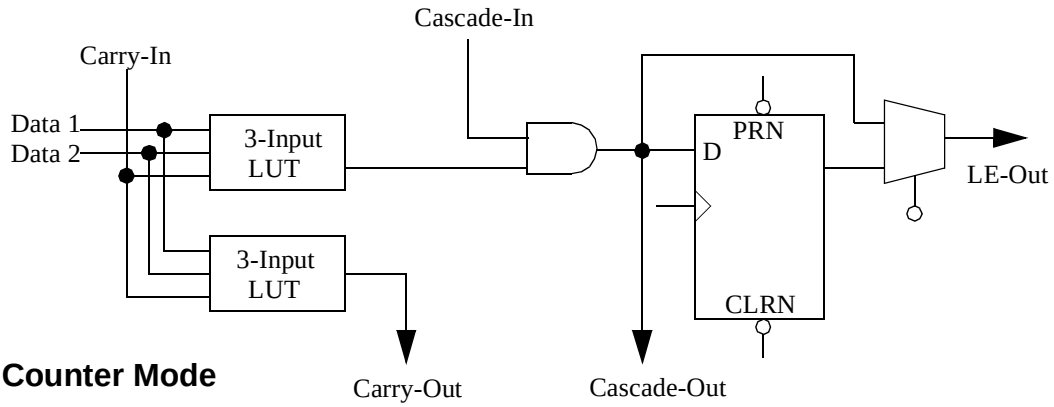
- LEs can operate in a number of different modes, see Figure 10.

Department of Electrical
and Computer Engineering

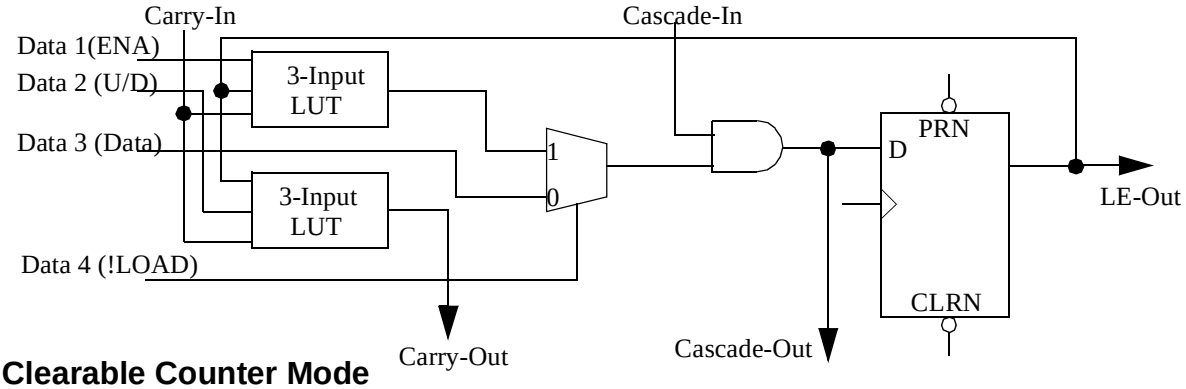
Normal Mode



Arithmetic Mode



Up/Down Counter Mode



Clearable Counter Mode

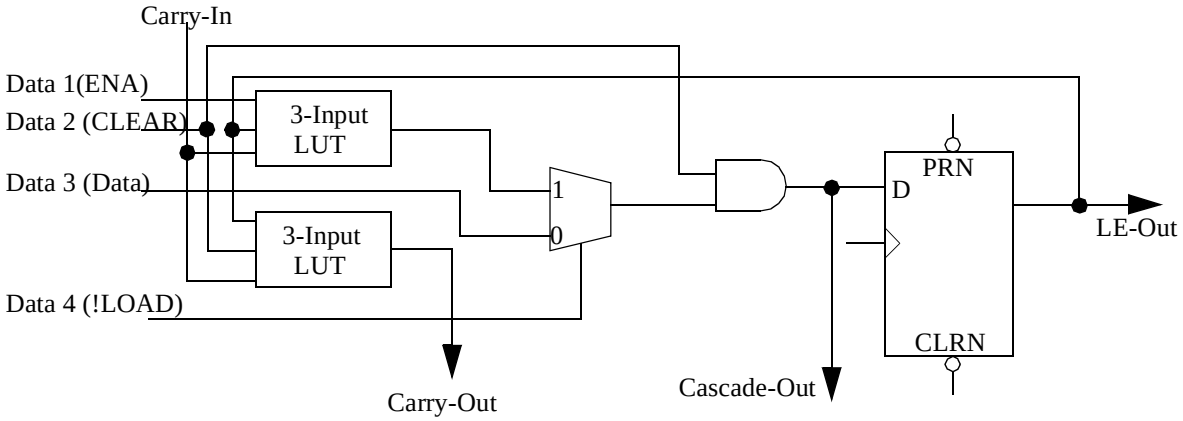


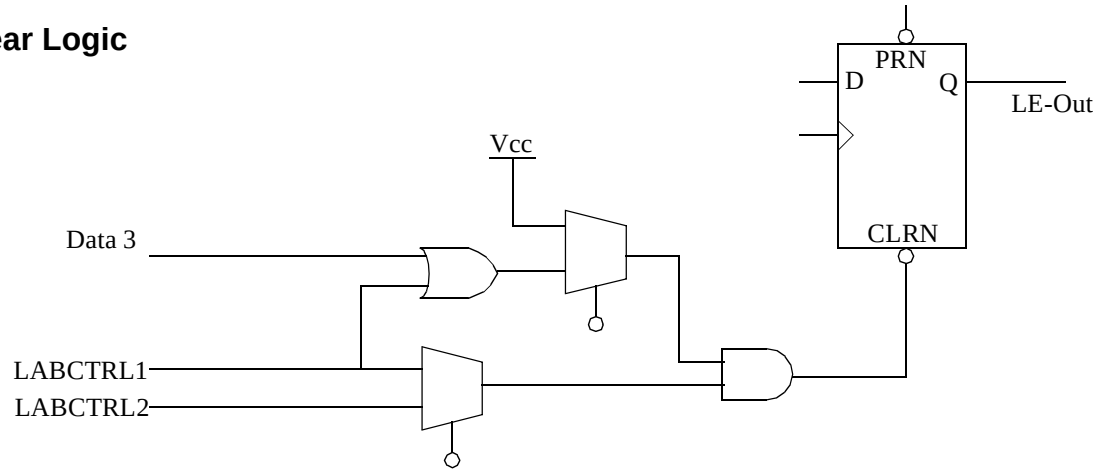
Figure 10 : Logic element operating modes

Department of Electrical
and Computer Engineering

- Clear and preset logic control - compiler selects the best control signal implementation during compilation.

- Preset can be provided by using Clear and inverting the output of the register. Inversion control is available for the inputs to both the LEs and the IOEs.

Clear Logic



Preset Logic

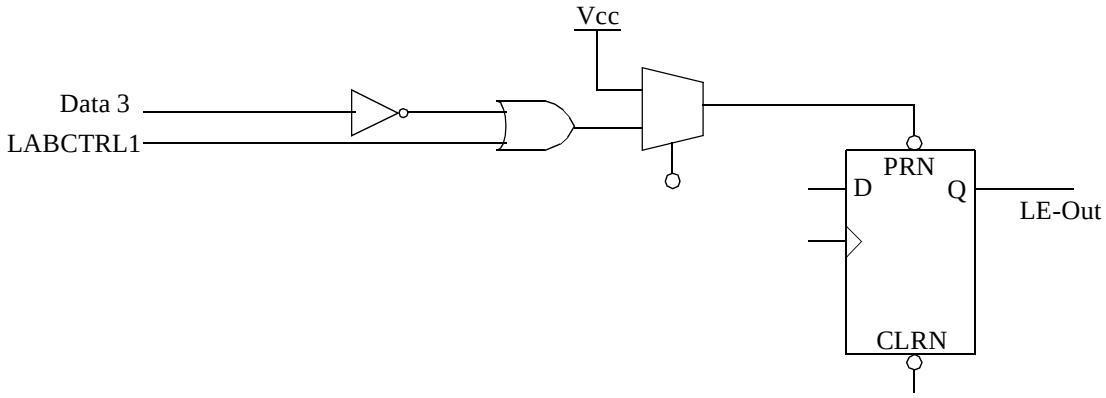


Figure 11 : Logic Element Clear and Preset Logic

Logic Array Block

- Logic Array Block (LAB) consists of eight LEs, their associated carry and cascade chains, LAB control signals, and the LAB local interconnect.

- Provides the course grain structure of the architecture which allows efficient routing with high device utilization and performance.

- Each LAB provides four control signals that can be used in all eight LEs. Two can be used as clocks and two as preset and clear. Programmable inversion is available on these signals. These signals provide very low clock skew to all the elements of the array.

- See Figure 12 for a block diagram of the logic array block.

- FastTrack Interconnect – connections between LEs and the device I/O pins. Provides fast and predictable interconnection delays (differs from FPGAs in this respect).

- See Figure 13 for the details of the row to column interconnect.

- Figure 14 shows the interconnect resources for the Flex 8000. Note that inputs into the LAB have to come via a row. Also the column resources are very scarce, and are their use is closely supervised by the compiler.

- Figure 15 shows the row to IOE connections. Inputs from an IOE can drive two separate row channels. When an IOE is used as an output the signal is driven by an n-to-1 multiplexer that selects the row channel. The size of the multiplexer varies with the number of columns in a device– e.g. some devices have 21-1 multiplexer, others a 27-1 multiplexer. Note that each pin has a limited row interconnect – cannot interconnect to an arbitrary row channel.

Department of Electrical
and Computer Engineering

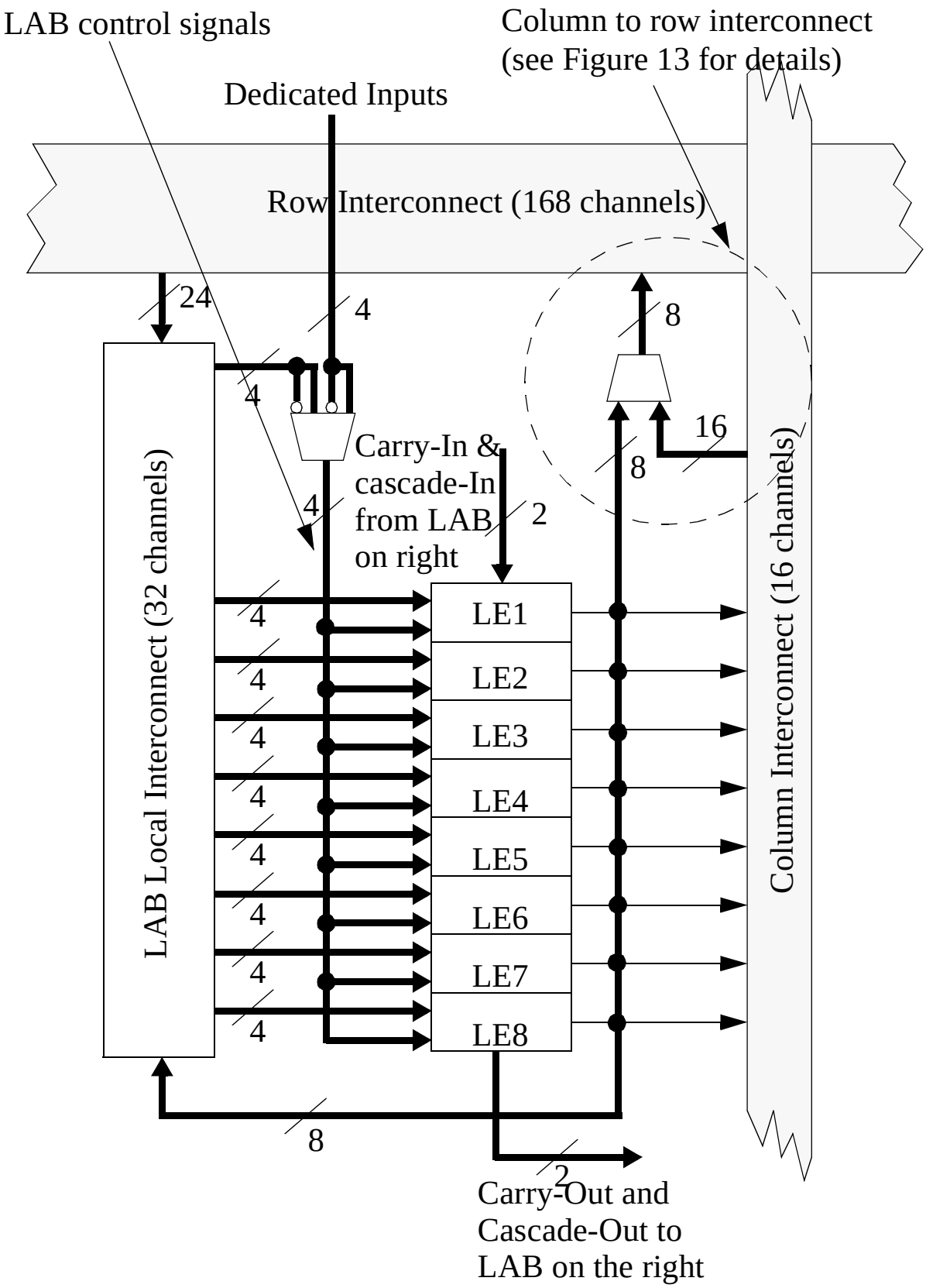


Figure 12 : Logic Array Block

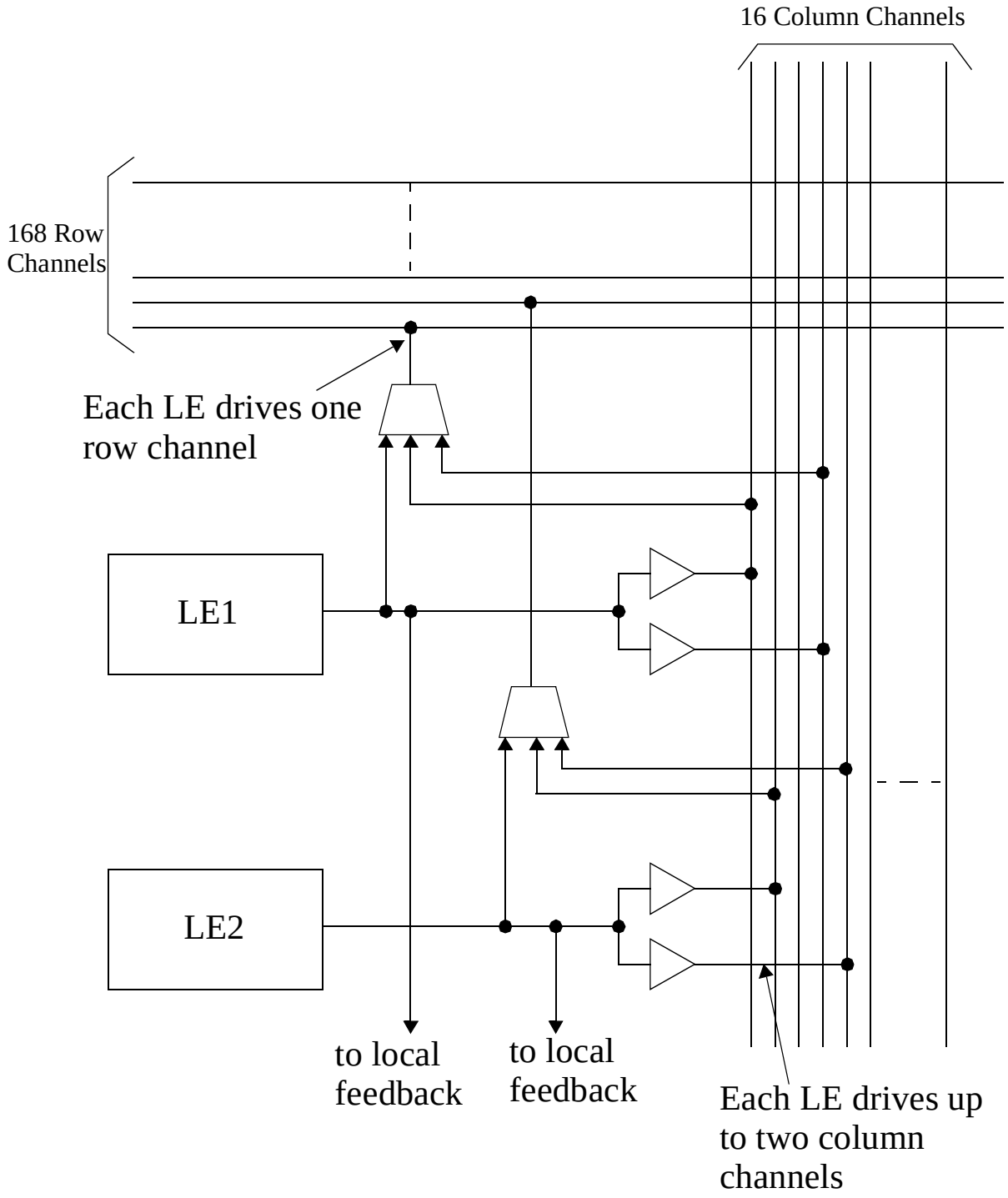


Figure 13 : LAB connections to row and column interconnect

Department of Electrical
and Computer Engineering

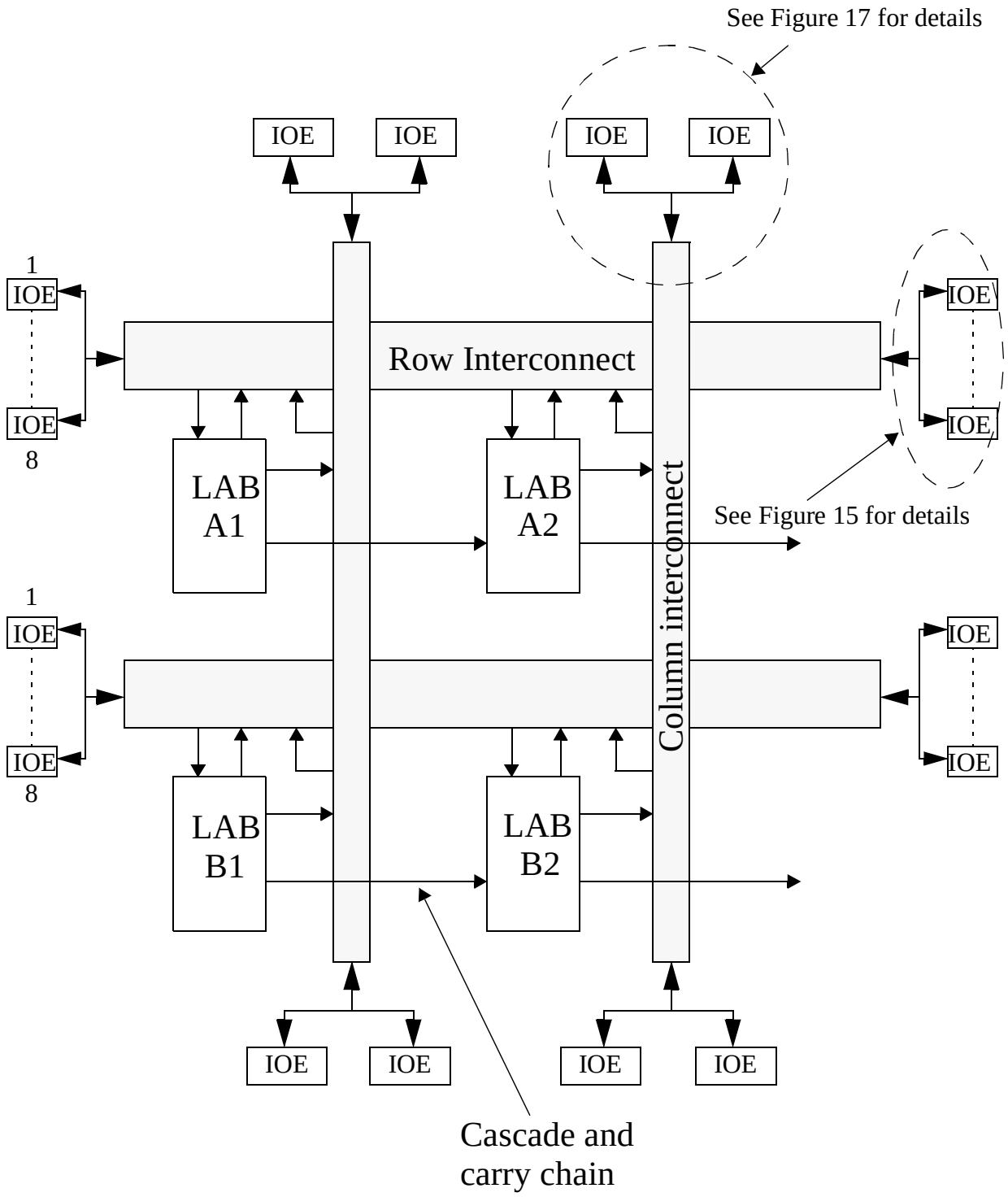


Figure 14 : Flex 8000 device interconnect resources

Department of Electrical
and Computer Engineering

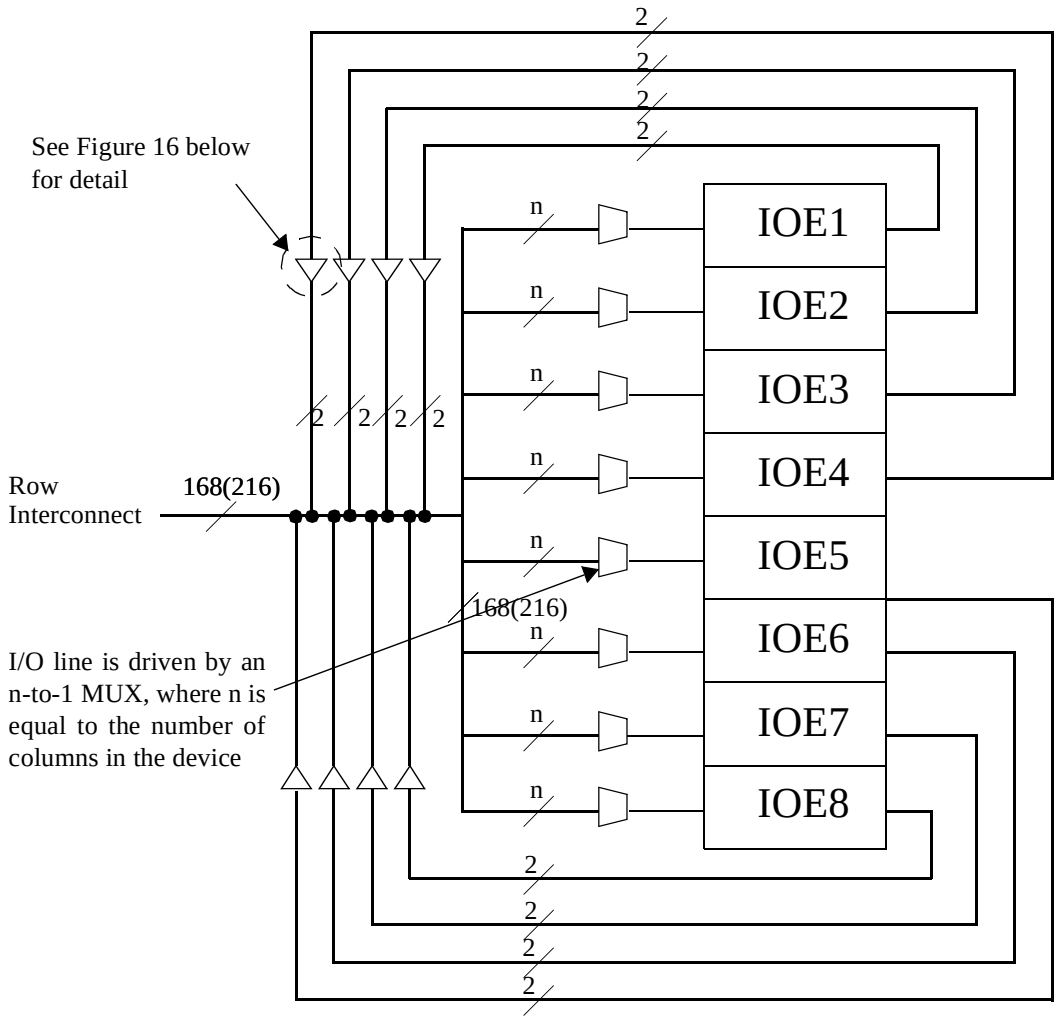


Figure 15 : Flex 8000 row to IOE connection

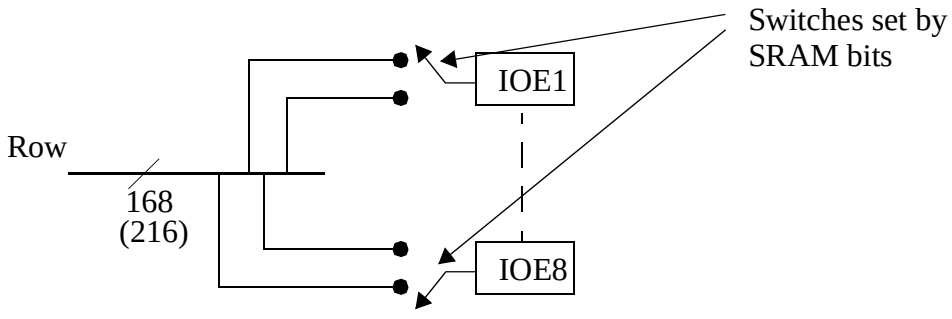


Figure 16 : Detail of the IOE to row interconnect selection

- There are IOEs on the top and bottom of each column I/O interconnect.

- The output of an 8-to-1 multiplexer is connected to each of the IOEs. See Figure 17.

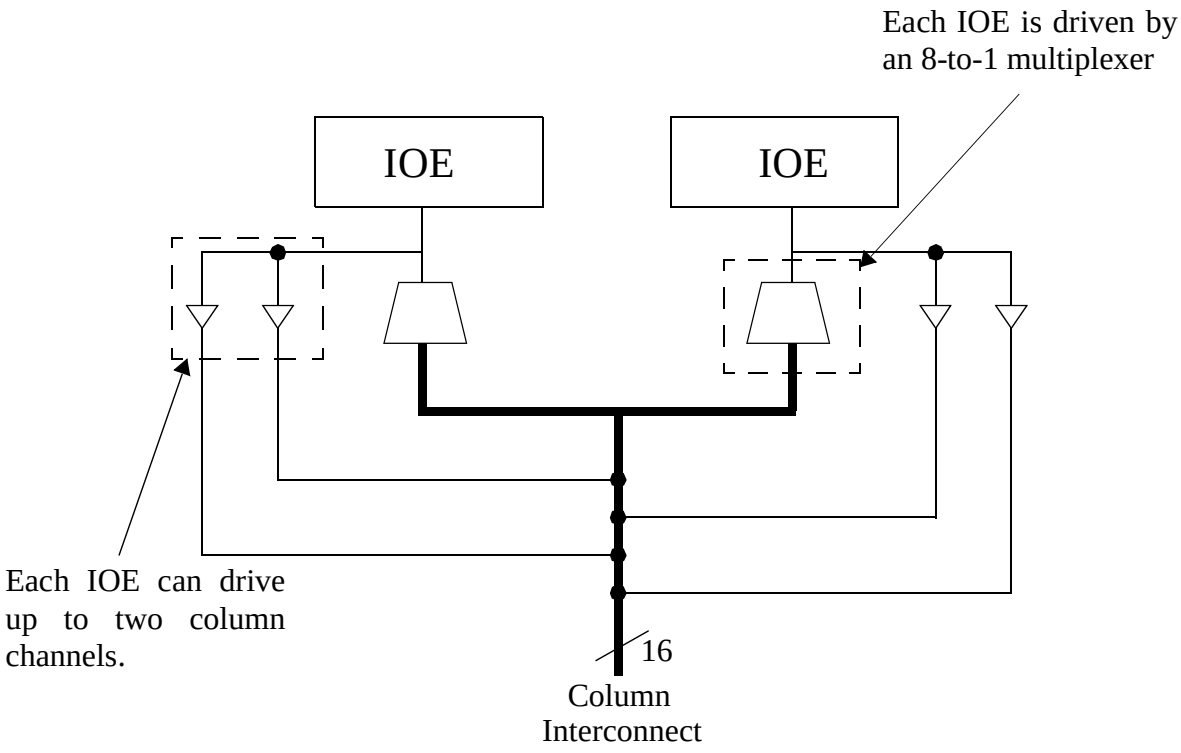


Figure 17 : Flex 8000 column to IOE connection

- The Input-Output Elements (IOEs) themselves are a reasonably complex circuit. They can be used for input, output or bidirectional I/O (See Figure 18).

- They are present at the ends of the row and column interconnects.

- Each I/O pin has a register that can be used for latching input data or output data.

- Output buffers have an adjustable slew rate control so that one can trade noise off against speed.

- The IOE can be controlled (i.e. clearing of the register, clocking of the register) via a dedicated “peripheral bus” that skirts the whole chip.

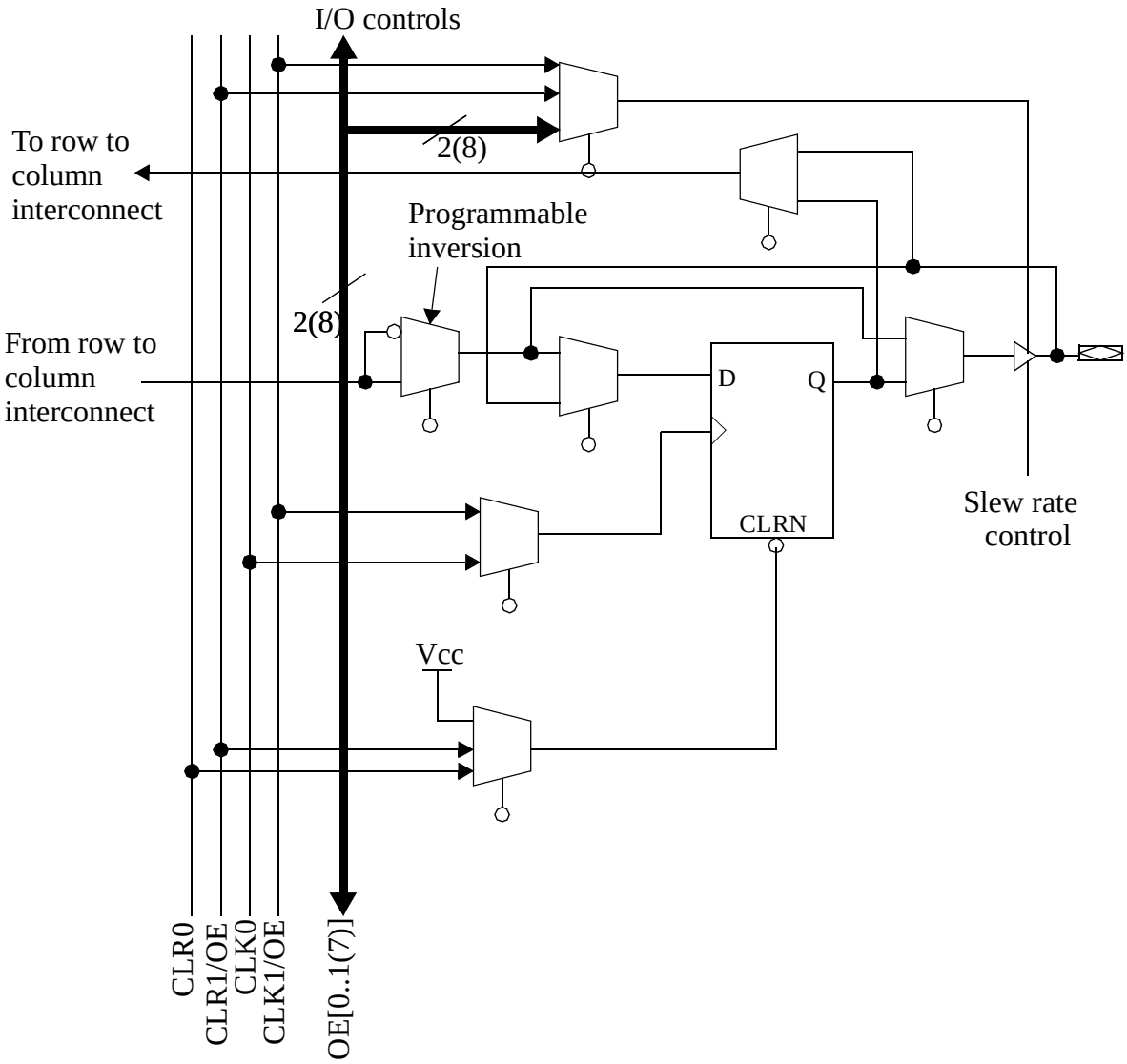


Figure 18 : I/O element (IOE) for a typical Altera EPLD.

- Signals for the peripheral bus can be generated by any of the 4 dedicated input lines, or signals on the row interconnect channels (the number of row channels corresponds to the number of columns in the device (i.e. 13, 21 or 27 depending on the device). See Figure 19

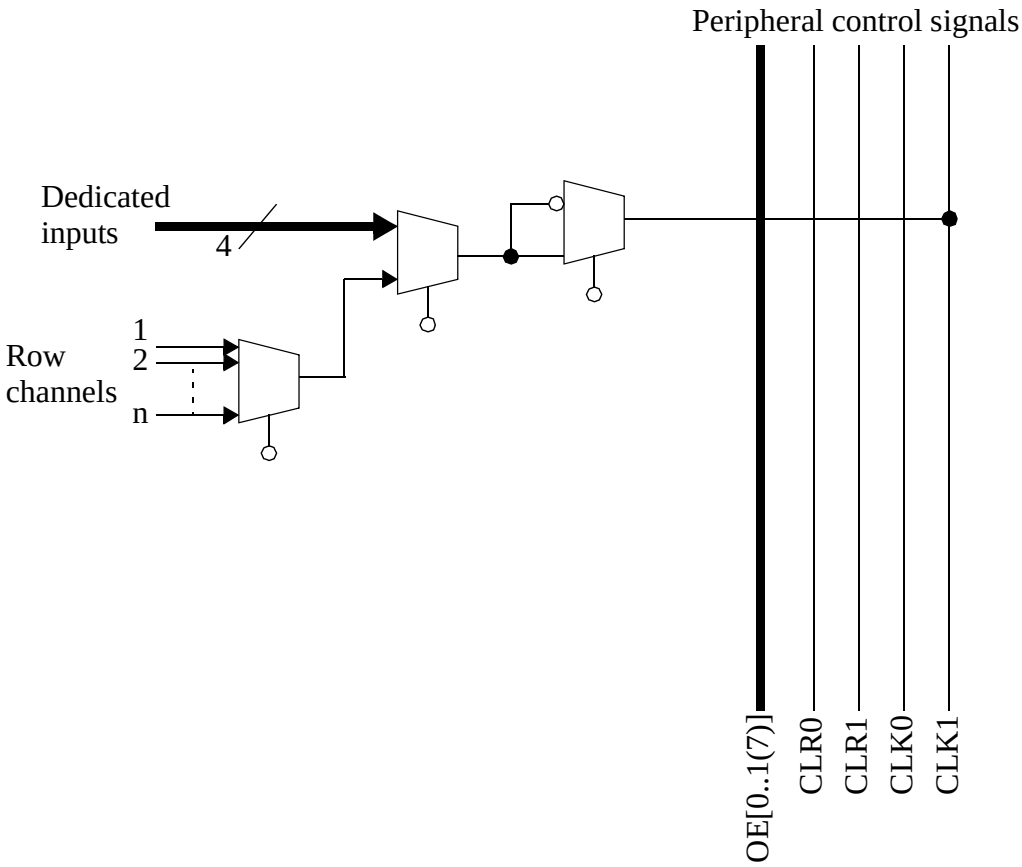


Figure 19 : Flex 8000 peripheral control bus

Department of Electrical
and Computer Engineering

FPGA (Field Programmable Gate Array) Devices

- In many ways these devices are similar to the EPLD based devices.
- Main difference is the routing. The FPGA devices have a more flexible routing system at the expense of having more routing delays.
- Modern FPGAs are still based on the use of logic blocks -Figure 20.

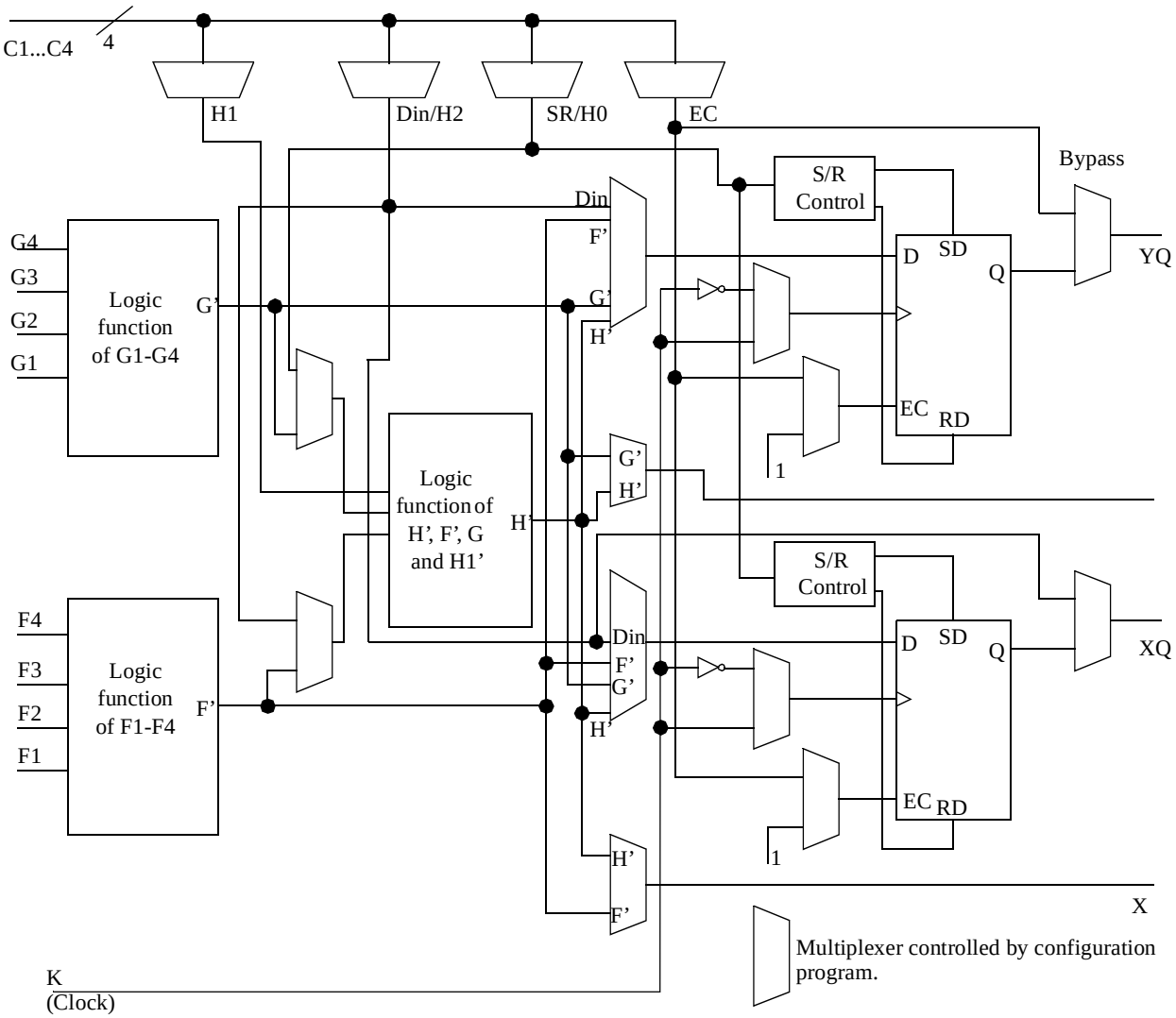


Figure 20 : Logic cell in XC4000 Xilinx FPGA.

Other features:

- The logic function blocks are implemented using RAM (as in Altera), but they can also be configured as RAM cells for general use.

- Have I/O Blocks (similar to the Altera) which connect to the pins and allow input and output into and out of the chip.

- Has internal tristate buffers so one can build tristate buses on the chip. Also has programmable pull-up resistors which allow wired AND functions.

- The major difference between the FPGA and the CPLD (Complex Programmable Logic Device) are the routing options available in the former. See Figure 21.

- Some nomenclature: Single means the lines run from one CLB (Complex Logic Block) to the next and there is a connection block at each CLB, double means that there is a connection block every second CLB, quad means that there is a connection block every fourth CLB.

- The connection blocks allow any line to be connected to any other line. This is implemented via a switching matrix. Figure 22 shows a conceptual view of the interconnect for the Xilinx FPGAs.

- The switch matrix is implemented as a set of six pass transistors. The quad line interconnect have a switch matrix together with a buffer (provides a high speed line). Long lines (run the whole length and width of the chip) can be driven by tristate buffers to form chip wide buffers. See Figure 23.

Department of Electrical
and Computer Engineering

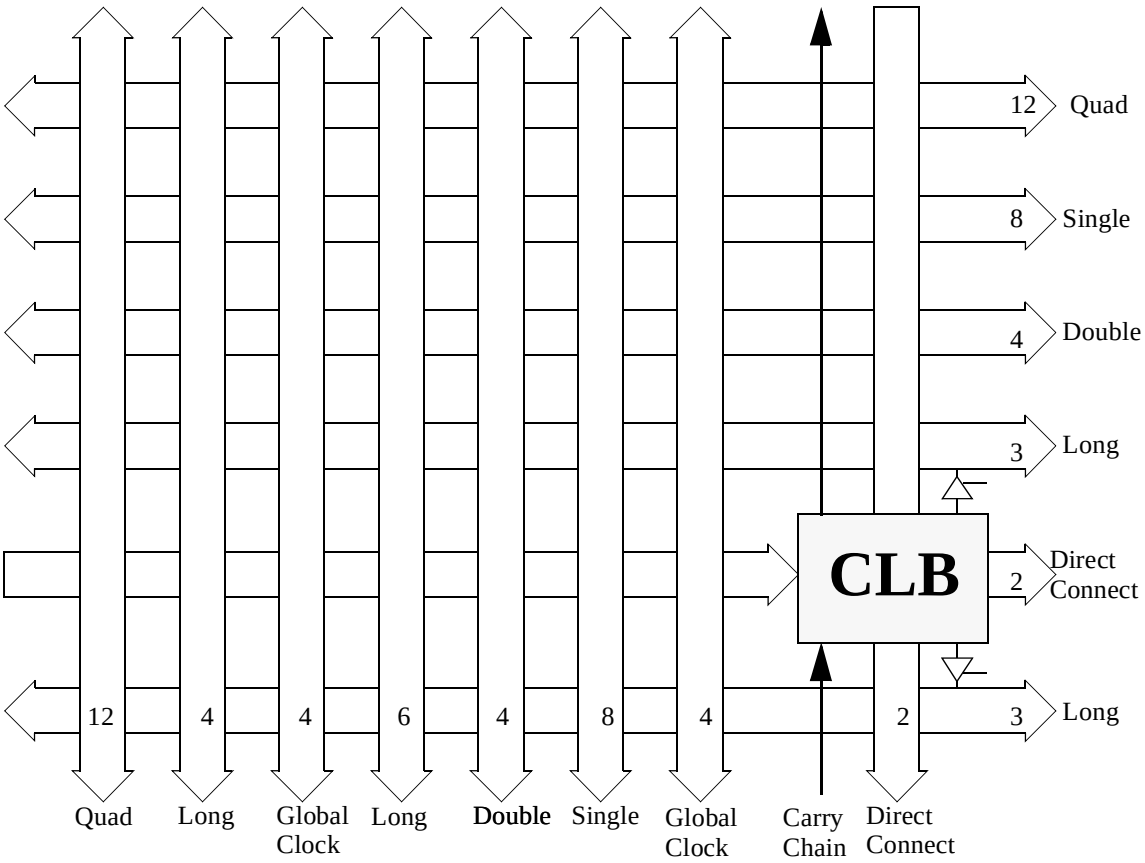


Figure 21 : High-level routing diagram of the XC4000EX series CLB

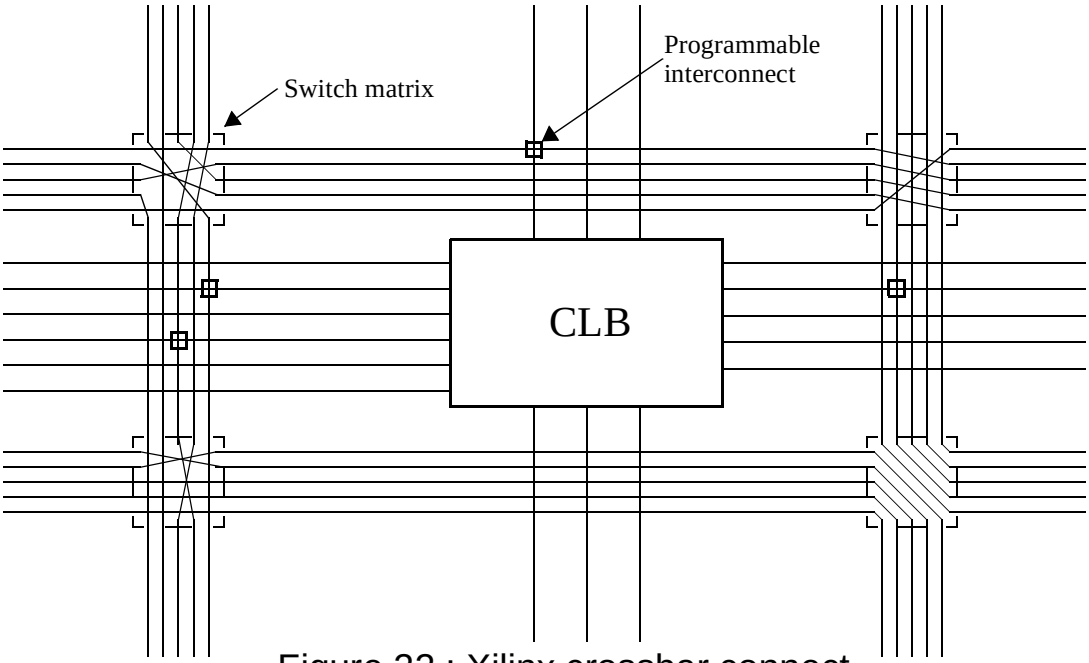


Figure 22 : Xilinx crossbar connect

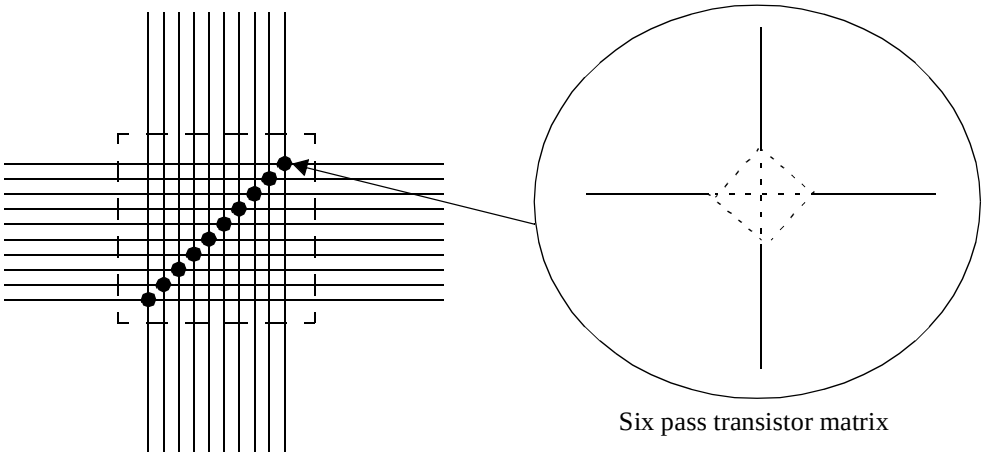


Figure 23 : Programmable switch matrix