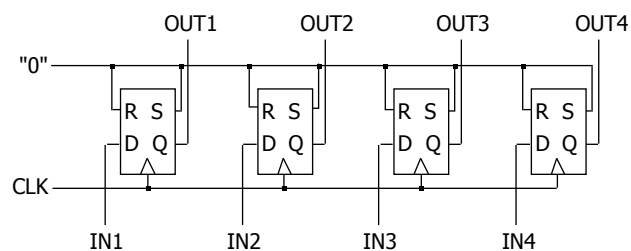


## Ensembles of flip-flops

- Registers
- Shift registers
- Counters

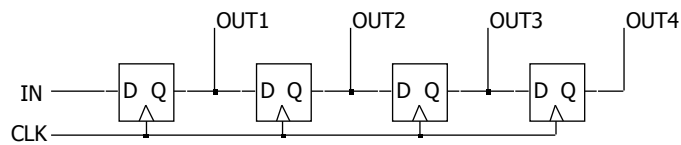
## Registers

- Collections of flip-flops with similar controls and logic
  - stored values somehow related (for example, form binary value)
  - share clock, reset, and set lines
  - similar logic at each stage
- Examples
  - shift registers
  - counters



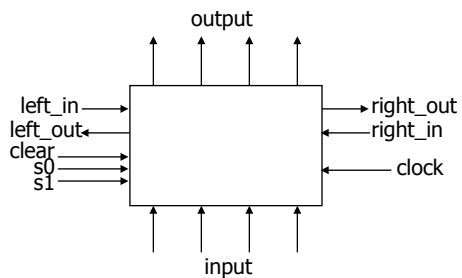
## Shift register

- Holds samples of input
  - store last 4 input values in sequence
  - 4-bit shift register:



## Universal shift register

- Holds 4 values
  - serial or parallel inputs
  - serial or parallel outputs
  - permits shift left or right
  - shift in new values from left or right



clear sets the register contents and output to 0

s1 and s0 determine the shift function

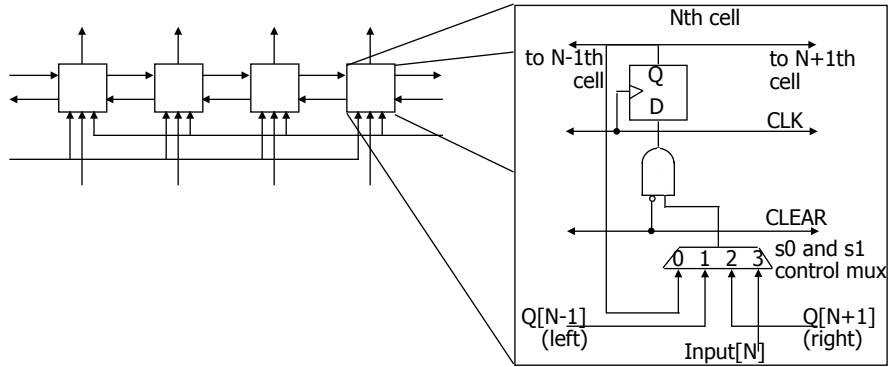
s0	s1	function
0	0	hold state
0	1	shift right
1	0	shift left
1	1	load new input

## Design of universal shift register

- Consider one of the 4 flip-flops

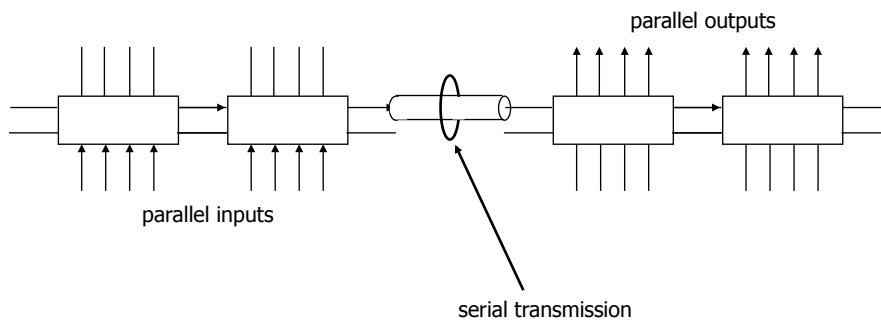
- new value at next clock cycle:

clear	s0	s1	new value
1	-	-	0
0	0	0	output
0	0	1	output value of FF to left (shift right)
0	1	0	output value of FF to right (shift left)
0	1	1	input



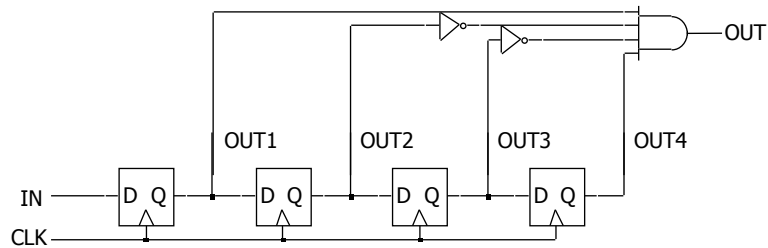
## Shift register application

- Parallel-to-serial conversion for serial transmission



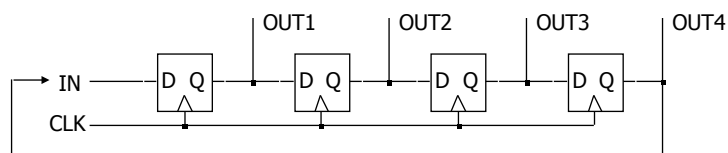
## Pattern recognizer

- Combinational function of input samples
  - in this case, recognizing the pattern 1001 on the single input signal



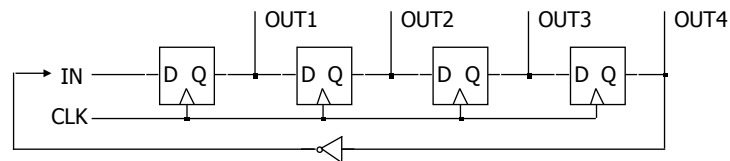
## Counters

- Sequences through a fixed set of patterns
  - in this case, 1000, 0100, 0010, 0001
  - if one of the patterns is its initial state (by loading or set/reset)



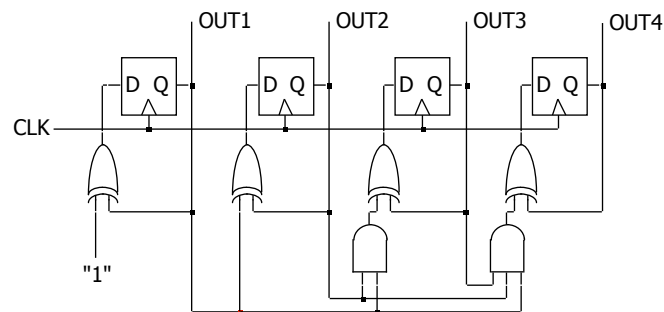
## Activity

- How does this counter work (assuming it starts in state 0000)?



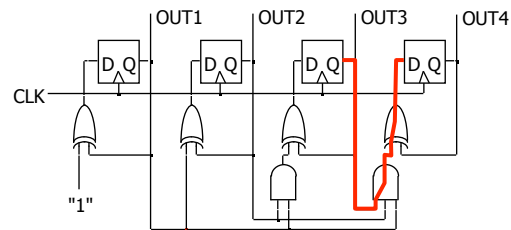
## Binary counter

- Logic between registers (not just multiplexer)
  - XOR decides when bit should be toggled
  - always for low-order bit,  
only when first bit is true for second bit,  
and so on



## How fast is our counter?

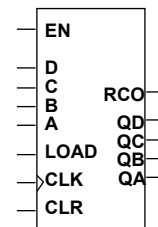
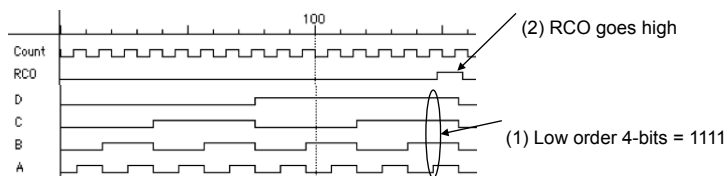
- $\text{Period} > T_{\text{propFF}} + T_{\text{propCL}} + T_{\text{setupFF}}$



- $\text{Period} > 3.6\text{ns} + \max(T_{\text{propCL}}) \{\text{over all paths from Qs to Ds}\} + 1.8\text{ns}$
- $\text{Period} > 5.4\text{ns} + T_{\text{propXOR}} + T_{\text{propAND3}}$
- $\text{Frequency} < 1/\text{Period}$

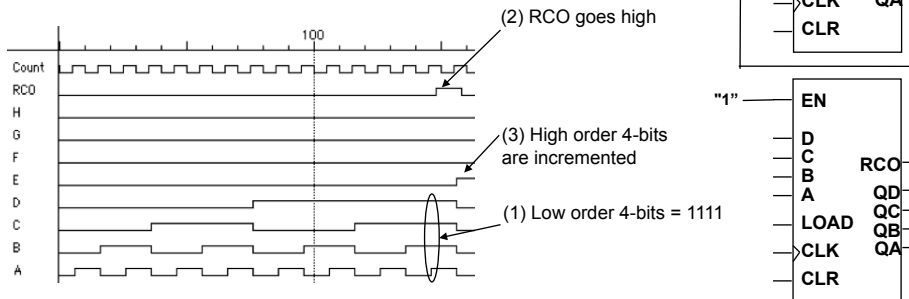
## Four-bit binary synchronous up-counter

- Standard component with many applications
  - positive edge-triggered FFs w/ synchronous load and clear inputs
  - parallel load data from D, C, B, A
  - enable inputs: must be asserted to enable counting
  - RCO: ripple-carry out used for cascading counters
    - high when counter is in its highest state 1111
    - implemented using an AND gate



## Eight-bit counter

- Use of enable to allow high-order four bits to count
  - Always enable 4 low-order bits
  - Enable 4 high-order bits only when RCO is high for low-order bits (reached 15)



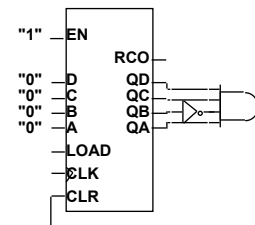
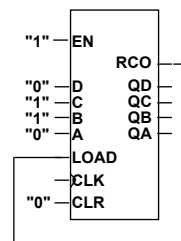
Autumn 2010

CSE370 - XV - Registers and Counters

13

## Offset counters

- Starting offset counters
  - use of synchronous load
  - e.g., 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1111, 0110, ...
- Ending offset counter
  - comparator for ending value
  - e.g., 0000, 0001, 0010, ..., 1100, 1101, 0000
- Combinations of the above (start and stop value)
- Speed?



Autumn 2010

CSE370 - XV - Registers and Counters

14

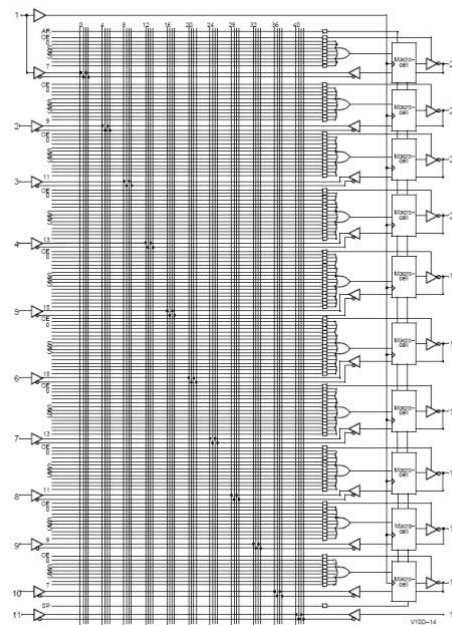
## Sequential logic and programmable logic

- FFs added to PLAs, PALs, and ROMs
- FFs added to FPGAs
- Other features
  - Feedback of Qs back in as inputs
  - Multiplexor for selecting combinational or sequential function
  - Arithmetic mode in FPGAs
  - Many other options .....

## The 22V10 PAL

- Combinational logic elements (SoP)
- Sequential logic elements (D-FFs)
- Up to 10 outputs
- Up to 10 FFs
- Up to 22 inputs
  
- A typical PAL

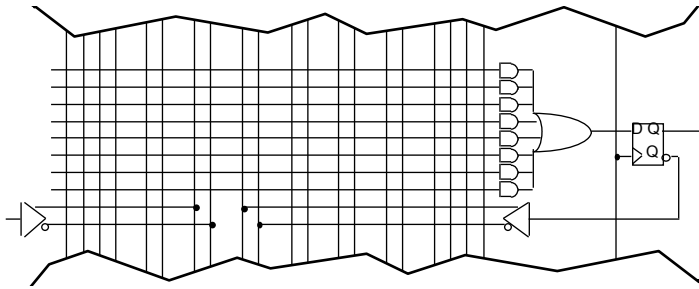
Functional Logic Diagram for PALC22V10D





## Implementation using PALs

- Programmable logic building block for sequential logic
  - macro-cell: FF + logic
    - D-FF
    - two-level logic capability like PAL (e.g., 8 product terms)



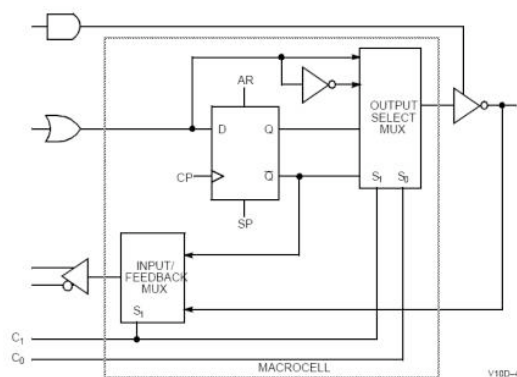
Autumn 2010

CSE370 - XV - Registers and Counters

17

## 22V10 PAL Macro Cell

- Sequential logic element + output/input selection

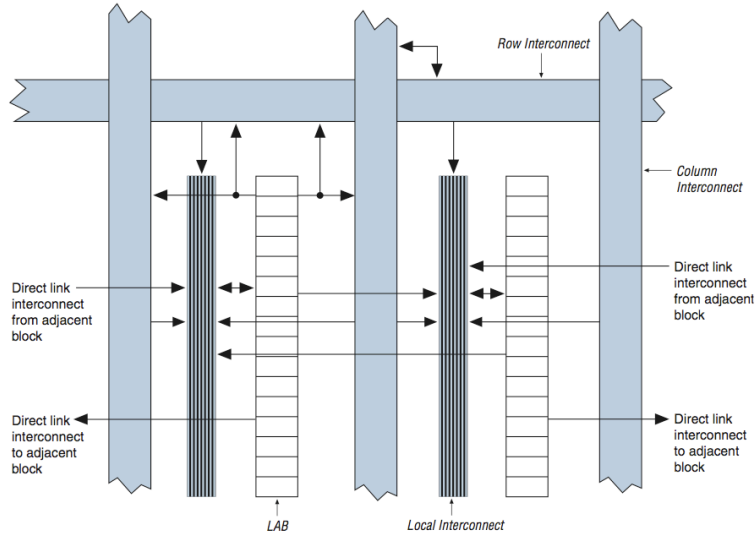


Autumn 2010

CSE370 - XV - Registers and Counters

18

## Cyclone II FPGA Logic Array Blocks (LABs)

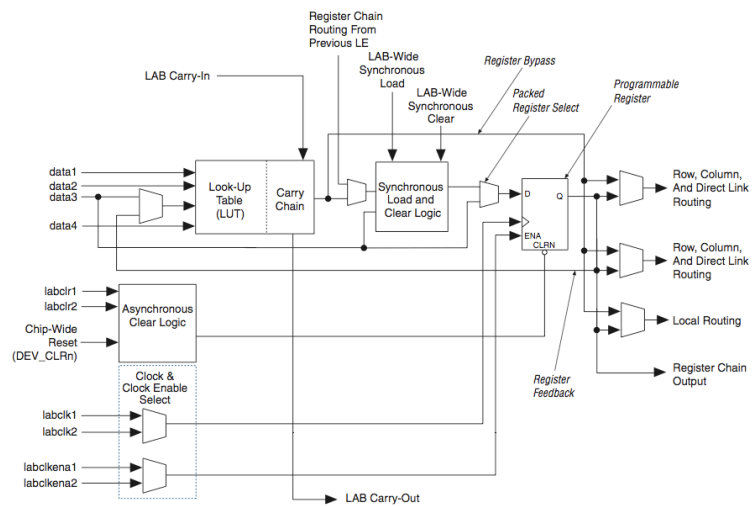


Autumn 2010

CSE370 - XV - Registers and Counters

19

## Cyclone II FPGA Logic Element (LE) (16 per LAB)

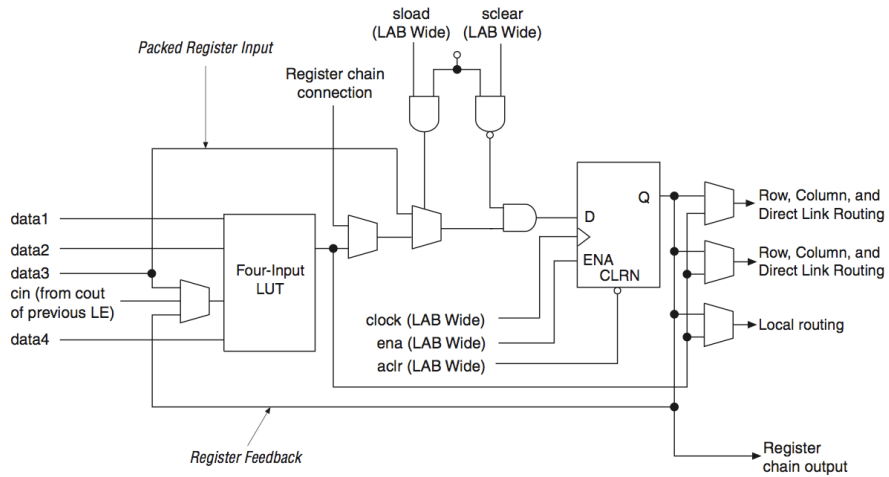


Autumn 2010

CSE370 - XV - Registers and Counters

20

## Logic Element (LE) in Normal Mode

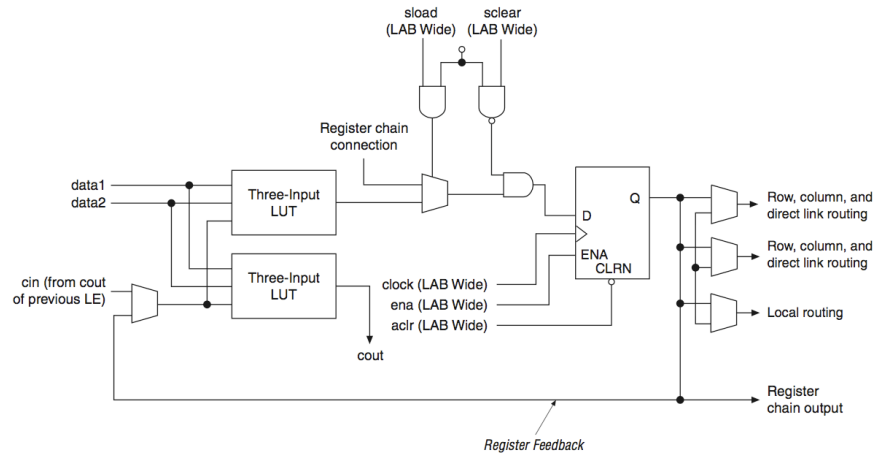


Autumn 2010

CSE370 - XV - Registers and Counters

21

## Logic Element (LE) in Arithmetic Mode



Autumn 2010

CSE370 - XV - Registers and Counters

22