CS294-7: Reconfigurable Computing

Day 20: Thursday, April 3

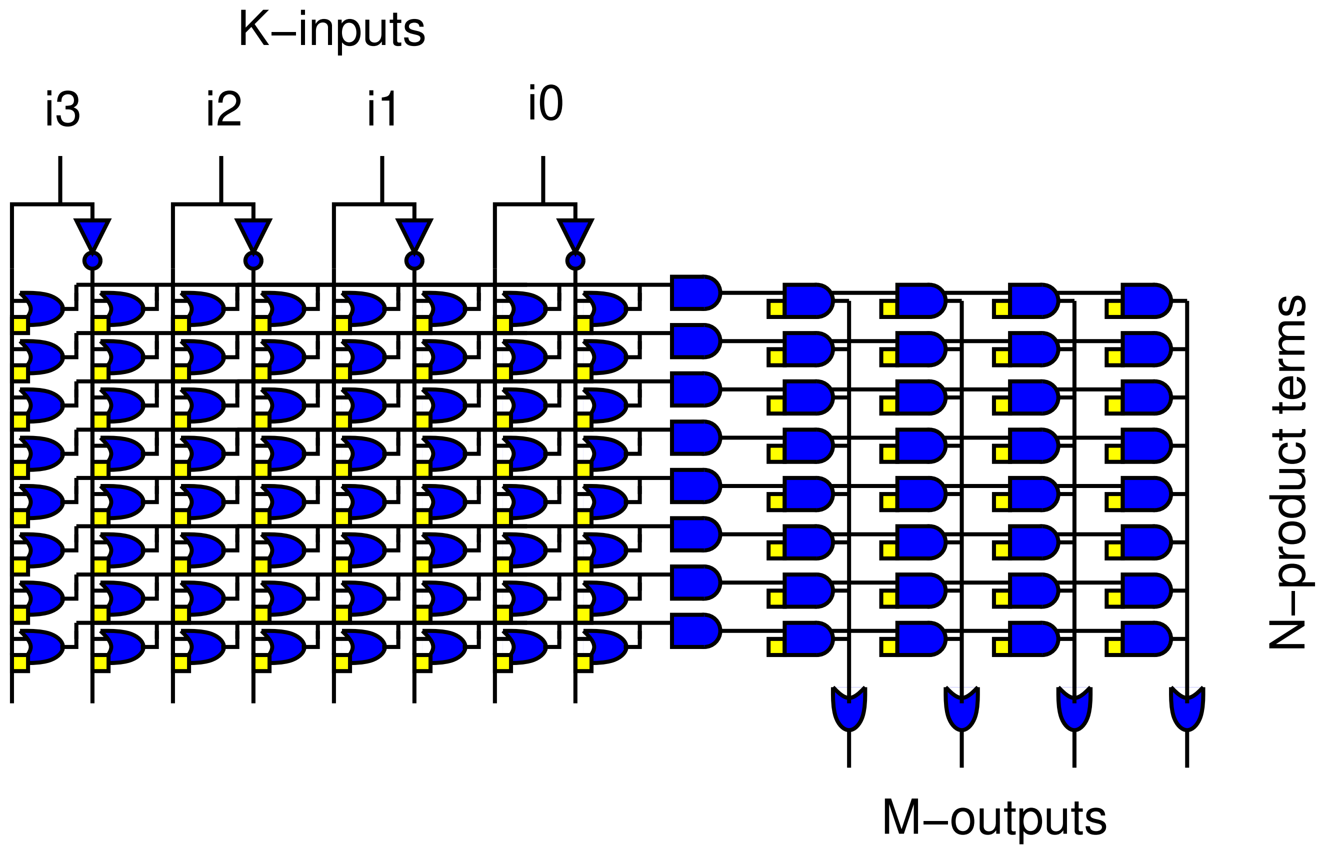
Compute Blocks

- PLAs
- ULMs

Altera component diagrams from Altera's datasheets.

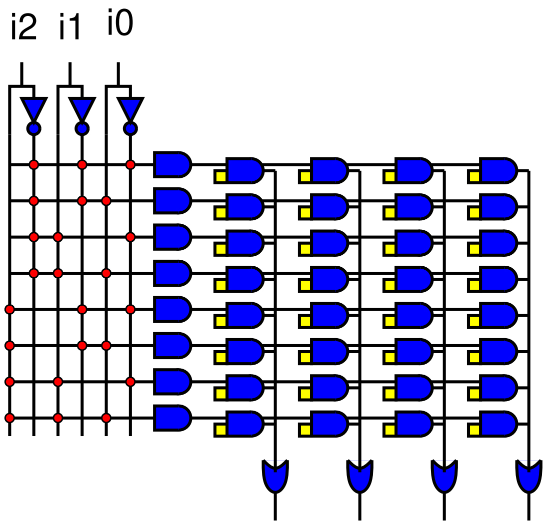
Programmable Logic Array

- PLA – early “programmable” array logic
- Two-level sum-of-products implementation

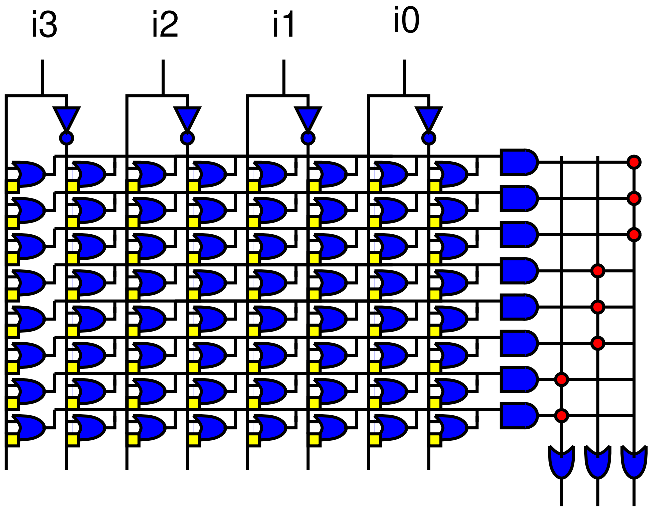


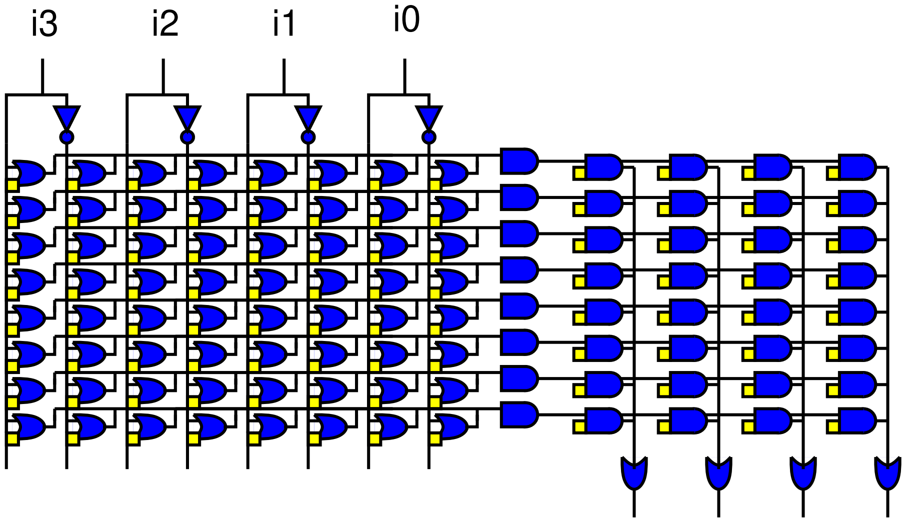
Array Logic

Memory (LUT)



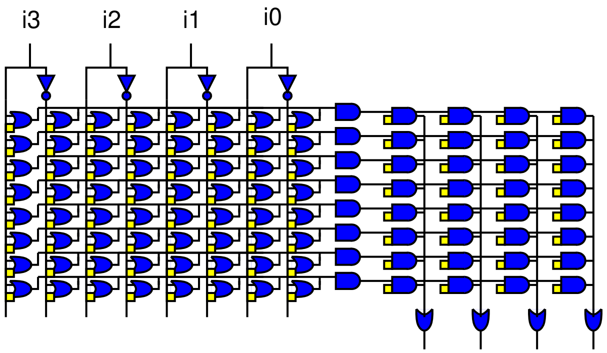
PAL





Programmable Logic Array

- Direct implementation of 2-level logic
- Fast implementation of large P-terms
- Number of P-terms in arbitrary function
  can be exponential in number of inputs
  (...and is for some common/interesting logic functions
  e.g. Parity, arithmetic)
- As with LUTs, can interconnect small PLAs with programmable interconnect



PLA Area v/s K and M

$$2 \leq K \leq 4 \quad M = 1$$
$$4 \leq K \leq 7 \quad M = 2$$
$$7 \leq K \quad M = 3$$

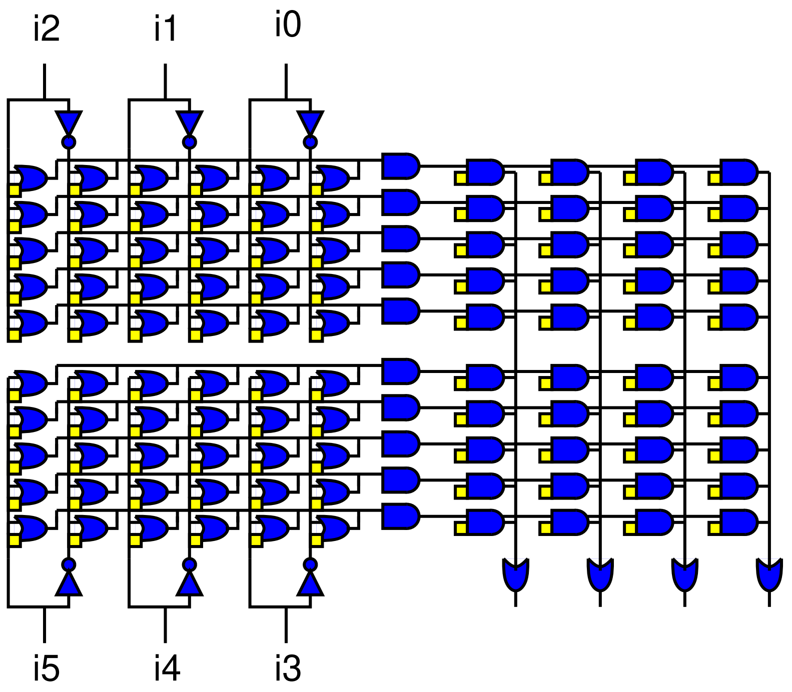
Smallest: $K = 10$, $M = 3$, $N = 12$

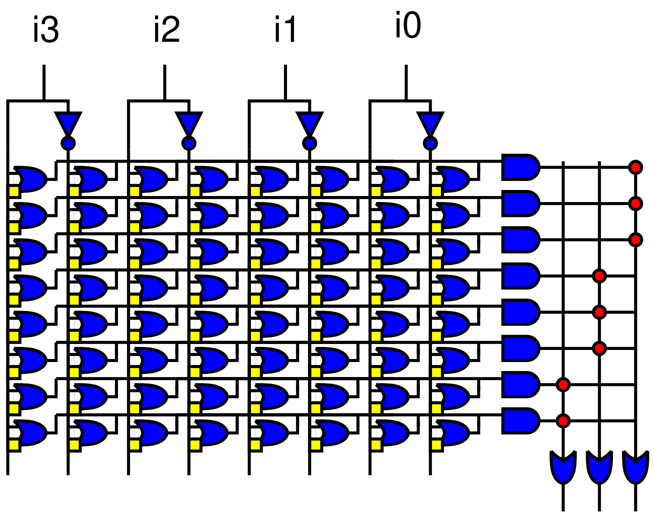
PLA v/s LUTs

- $A_{PLA=10.3.2}$ = 80–130% A_{4LUT}
  (300% on ECC benchmarks)
- Claim: 40% fewer logic levels?

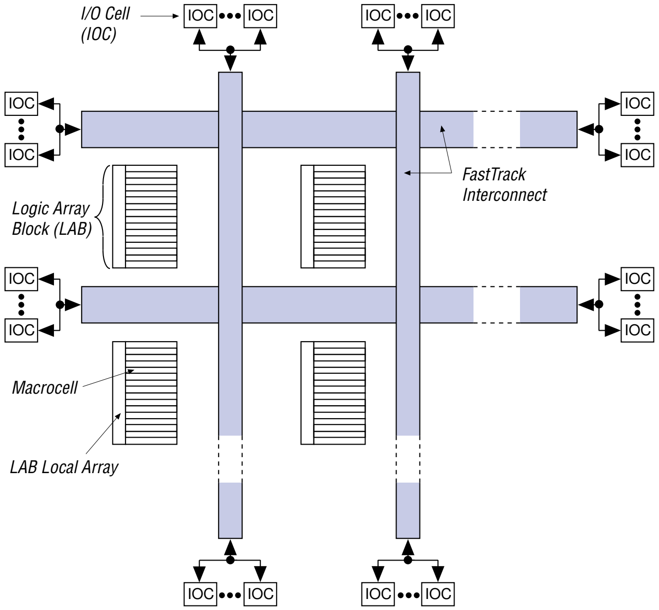
Areas to Optimize PLA

- input folding
- fixed OR-plane

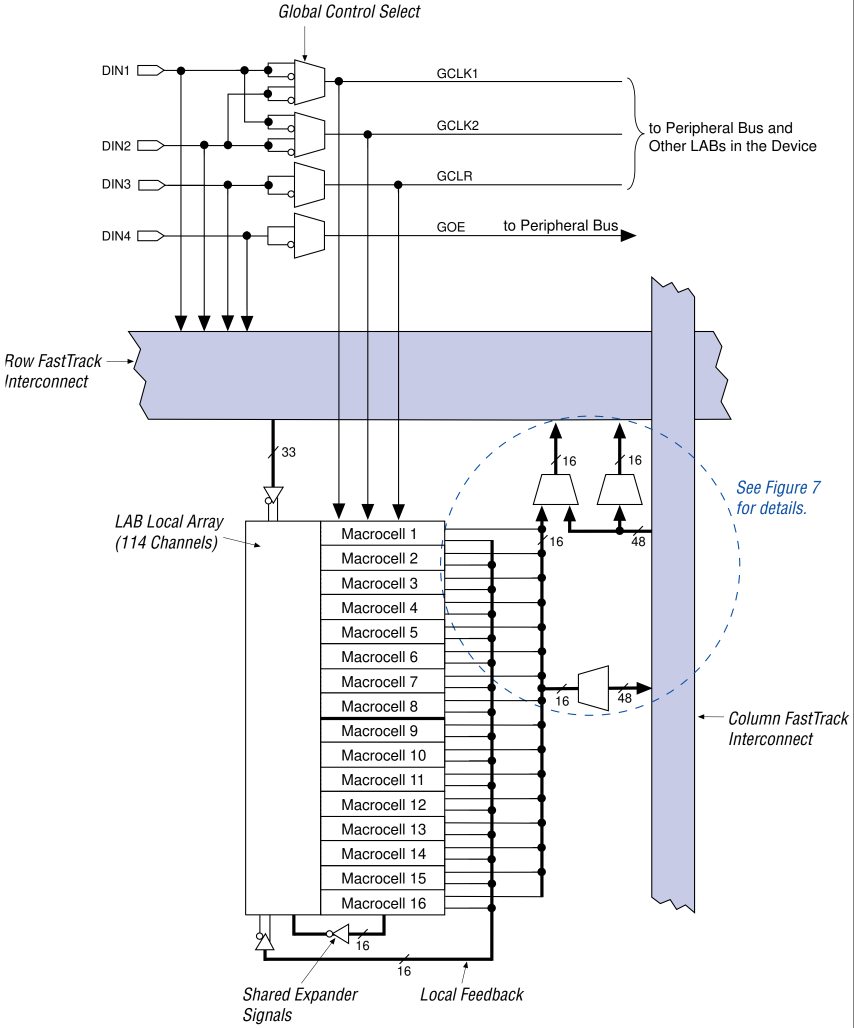




Altera 9000

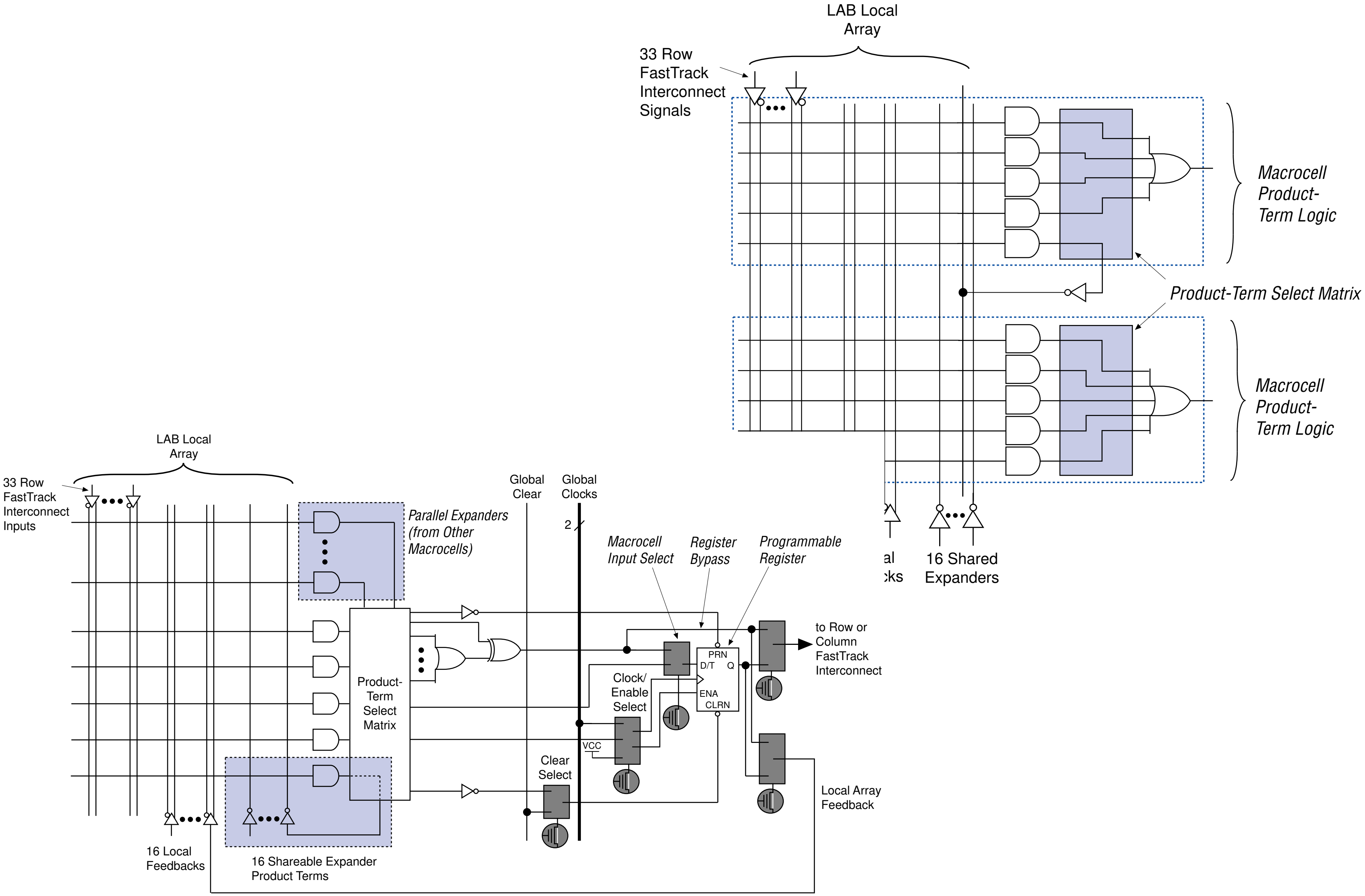


I/O Cell
(IOC)
IOC
Logic Array
Block (LAB)
FastTrack
Interconnect
Macrocell
LAB Local Array



Global Control Select
DIN1
DIN2
DIN3
DIN4
GCLK1
GCLK2
GCLR
GOE
to Peripheral Bus and
Other LABs in the Device
to Peripheral Bus
Row FastTrack
Interconnect
33
16
16
See Figure 7
for details.
LAB Local Array
(114 Channels)
Macrocell 1
Macrocell 2
Macrocell 3
Macrocell 4
Macrocell 5
Macrocell 6
Macrocell 7
Macrocell 8
Macrocell 9
Macrocell 10
Macrocell 11
Macrocell 12
Macrocell 13
Macrocell 14
Macrocell 15
Macrocell 16
48
Column FastTrack
Interconnect
Shared Expander
Signals
Local Feedback

Altera 9000



LAB Local
Array
33 Row
FastTrack
Interconnect
Signals
Macrocell
Product-
Term Logic
Product-Term Select Matrix
Macrocell
Product-
Term Logic
16 Shared
Expanders
LAB Local
Array
33 Row
FastTrack
Interconnect
Inputs
Parallel Expanders
(from Other
Macrocells)
Global
Clear
Global
Clocks
2
Macrocell
Input Select
Register
Bypass
Programmable
Register
Product-
Term
Select
Matrix
PRN
D/T
Q
ENA
CLRN
Clock/
Enable
Select
VCC
Clear
Select
to Row or
Column
FastTrack
Interconnect
Local Array
Feedback
16 Local
Feedbacks
16 Shareable Expander
Product Terms

Universal Logic Modules

Logic Module

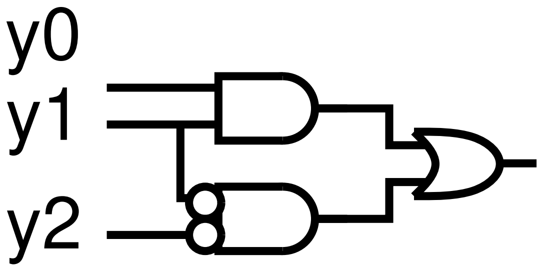
- M inputs
- implements any function of N ($N < M$) inputs
- Assume
  - get input and inversion (n)
  - inputs permutable (p)
  - output invertable (n)

Equivalence Classes

	n	np	npn
0			
1			
x_1	x_1	x_1	x_1
$\overline{x_1}$			
x_2	x_2		
$\overline{x_2}$			
$x_1 \cdot x_2$	$x_1 \cdot x_2$	$x_1 \cdot x_2$	$x_1 \cdot x_2$
$\overline{x_1} \cdot x_2$			
$\overline{x_2} \cdot x_1$			
$\overline{x_1} \cdot \overline{x_2}$			
$x_1 + x_2$	$x_1 + x_2$	$x_1 + x_2$	
$\overline{x_1} + x_2$			
$\overline{x_2} + x_1$			
$\overline{x_1} + \overline{x_2}$			
$x_1 \oplus x_2$	$x_1 \oplus x_2$	$x_1 \oplus x_2$	$x_1 \oplus x_2$
$\overline{x_1} \oplus x_2$			
16	5	4	3

Example ULM.2

	y_0	y_1	y_2
0	0	x_1	1
1	1	x_1	0
x_1	1	x_1	1
$\overline{x_1}$	0	x_1	0
x_2	x_2	x_1	$\overline{x_2}$
$\overline{x_2}$	$\overline{x_2}$	x_1	x_2
$x_1 \cdot x_2$	x_2	x_1	1
$\overline{x_1} \cdot x_2$	0	x_1	$\overline{x_2}$
$\overline{x_2} \cdot x_1$	$\overline{x_2}$	x_1	1
$\overline{x_1} \cdot \overline{x_2}$	0	x_1	x_2
$x_1 + x_2$	1	x_1	$\overline{x_2}$
$\overline{x_1} + x_2$	x_2	x_1	0
$\overline{x_2} + x_1$	1	x_1	x_2
$\overline{x_1} + \overline{x_2}$	$\overline{x_2}$	x_1	0
$x_1 \oplus x_2$	$\overline{x_2}$	x_1	$\overline{x_2}$
$\overline{x_1} \oplus x_2$	x_2	x_1	x_2

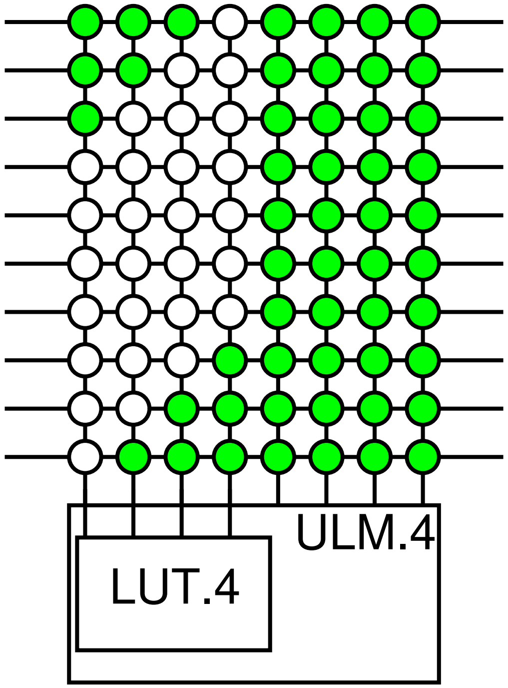


ULM

- ULMs always have more inputs than LUT
- Number of inputs in channel same
  - more switches
  - no more wires
- Can implement a restricted set of larger functions (which may need more channel wires to support)

ULM inputs and switches

n	Minimum ULM inputs
2	3
3	5
4	8
5	13
6	21



ULM-based LUT (1 of 2)

Can exploit these classes to build a restricted “LUT”

For 4-LUT:

- 208 npn-equivalent functions (8b)
- np-equivalent functions require 9b
- Zilic and Vranesic show practical 13b solution

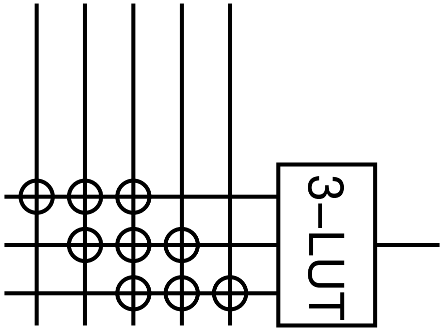
but...

ULM-based LUT (2 of 2)

...but, this requires input permutability in network:

- precludes $(K-1)W$ depopulation of input switches
- with dense encodings save 4-5 network bits/LUT using input permutability

$$\lim_{n \to \inf} \log_2 \left(\frac{n^k}{\binom{n}{k}} \right) \approx \lim_{n \to \inf} \log_2 \left(\frac{n^k}{\left(\frac{n^k}{k!}\right)} \right) = \log_2 (k!)$$



But, it's not a LUT...which might ease litigation... :-)

Summary

Looked at two alternative compute blocks:

- PLA (PAL)
  - Large PLA have similar problems to large LUTs
  - Can use PLA blocks + interconnect as with LUTs
  - Comparable density to LUTs (on most functions)
  - Possibly a speed advantage?
- ULM
  - Provide functions by signal inversion/permutation (all "programming" in network)
  - Bit savings in logic block (more than) compensated by increased interconnect cost