Topics

– Design Strategies

© John A. Chandy
Dept. of Electrical and Computer Engineering
University of Connecticut

Standard Cell Libraries

- How do you decide on the composition of the cell library?
  - Number of inputs
  - Transistor sizing for varying capacitive loads
  - Pullup/pulldown ratio

Dept. of Electrical and Computer Engineering
University of Connecticut

Standard Cell Libraries

AMI

AMERICAN MICROSYSTEMS, INC.

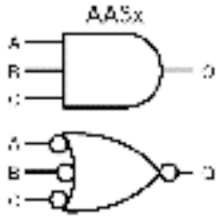
AA3x

AMI5HS 0.5 micron CMOS Standard Cell

Description

AA3x is a family of 3- nput gates which perform the logical AND function.

Logic Symbol



Truth Table

A	B	C	Q
L	X	X	L
X	L	X	L
X	X	L	L
H	H	H	H

HDL Syntax

Verilog ..... AA3x *inst_name* (Q, A, B, C);

VHDL ..... *inst_name*: AA3x port map (Q, A, B, C);

Dept. of Electrical and Computer Engineering
University of Connecticut

Standard Cell Libraries

Pin Loading

Pin Name	Equivalent Loads			
	AA31	AA32	AA34	AA36
A	1.0	1.0	2.0	2.9
B	1.0	1.0	1.9	2.9
C	1.0	1.0	1.9	2.9

Size And Power Characteristics

Cell	Equivalent Gates	Power Characteristics[a]	
		Static I_{DD} (T_J = 85°C) (nA)	EQL_{pxl} (Eq-load)
AA31	1.7	TBD	3.5
AA32	2.0	TBD	4.7
AA34	4.0	TBD	10.0
AA36	5.2	TBD	14.5

a. See page 2-13 for power equation.

Dept. of Electrical and Computer Engineering
University of Connecticut

Standard Cell Libraries

AA3x

AMERICAN MICRO SYSTEMS INC.

AMI5HS 0.5 micron CMOS Standard Cell

Propagation Delays (ns)

Conditions: T_J = 25°C, V_{DD} = 5.0V, Typical Process

Core Logic

AA31	Number of Equivalent Loads		1	3	6	10	13 (max)
	From: Any Input To: Q	t_{PLH} t_{PHL}	0.20 0.26	0.33 0.36	0.43 0.46	0.55 0.63	0.64 0.73
AA32	Number of Equivalent Loads		1	6	11	16	22 (max)
	From: Any Input To: Q	t_{PLH} t_{PHL}	0.20 0.28	0.39 0.42	0.46 0.52	0.55 0.61	0.63 0.71
AA34	Number of Equivalent Loads		1	10	20	30	40 (max)
	From: Any Input To: Q	t_{PLH} t_{PHL}	0.31 0.24	0.37 0.35	0.44 0.47	0.50 0.57	0.56 0.64
AA36	Number of Equivalent Loads		1	14	29	44	58 (max)
	From: Any Input To: Q	t_{PLH} t_{PHL}	0.26 0.25	0.36 0.36	0.44 0.44	0.51 0.53	0.57 0.61

Delay will vary with input conditions. See page 2-16 for interconnect estimates.

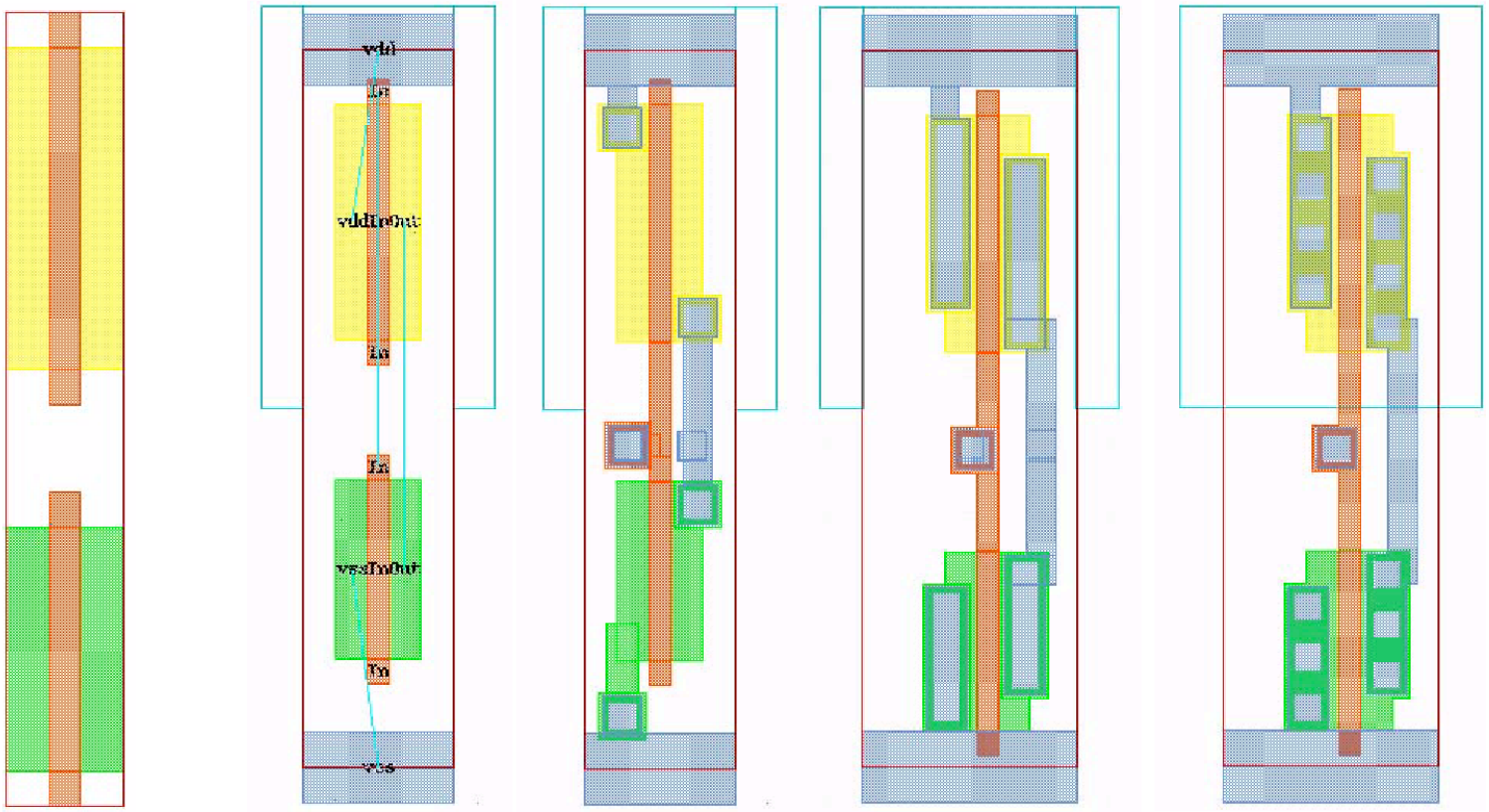
Dept. of Electrical and Computer Engineering
University of Connecticut

Compiled Cells

- Standard cells
  - Must be redesigned for every new process technology
  - Design options are limited because of discrete set of cells
- Customized cells would provide more flexibility
  - Automatic layout generation for design-specific requirements

Dept. of Electrical and Computer Engineering
University of Connecticut

Automatic Cell Generation



Initial transistor geometries | **Placed transistors** | **Routed cell** | **Compacted cell** | **Finished cell**

Courtesy Acadabra

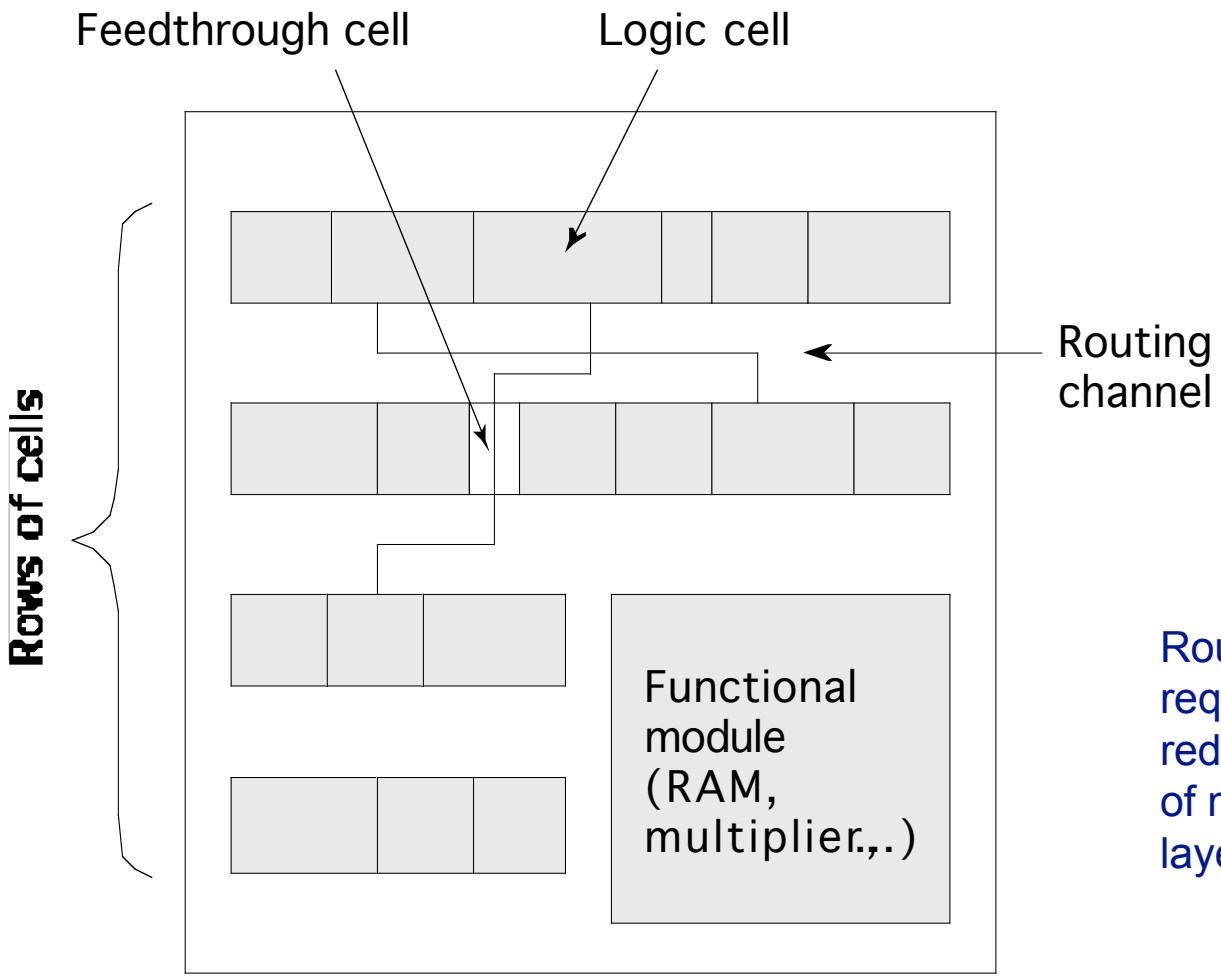
Dept. of Electrical and Computer Engineering
University of Connecticut

Macrocells

- Hard Macros
  - Predetermined physical design
  - Fixed transistor and wiring locations
  - Dense layout, optimized performance and power characteristics

Dept. of Electrical and Computer Engineering
University of Connecticut

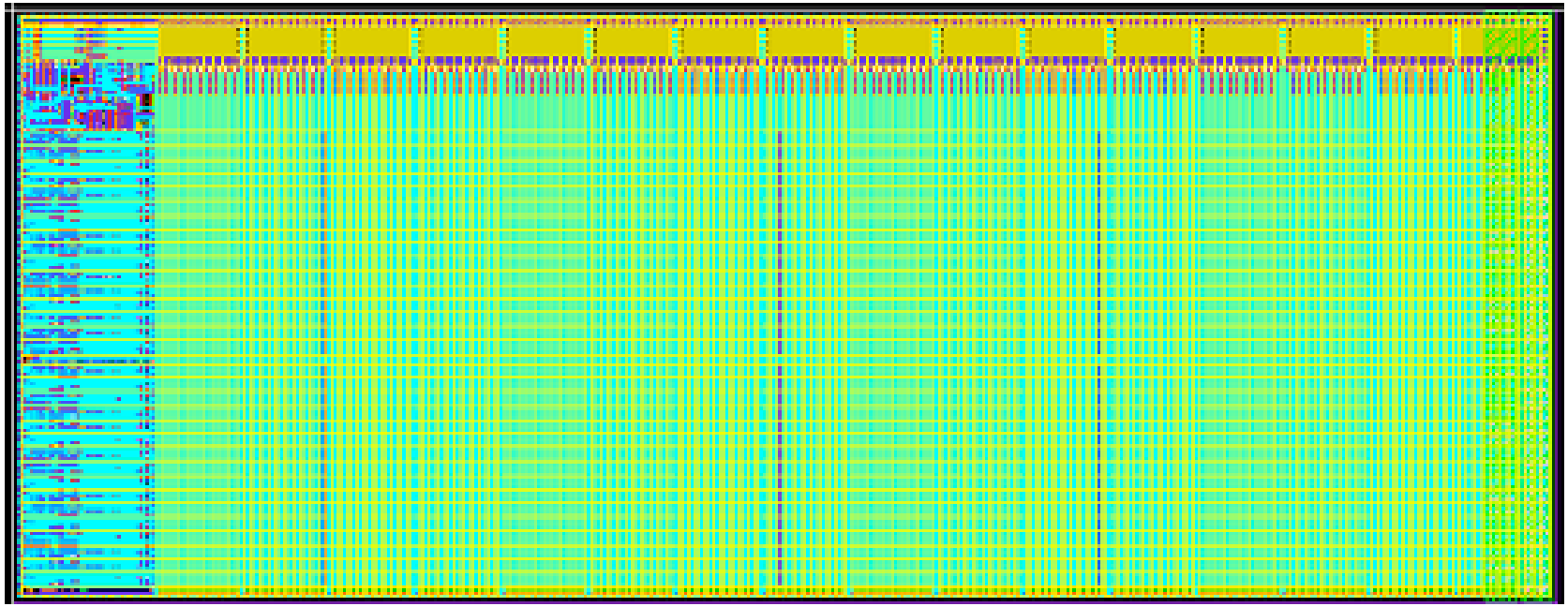
Cell-based Design (or standard cells)



Routing channel requirements are reduced by presence of more interconnect layers

Dept. of Electrical and Computer Engineering
University of Connecticut

Macrocells



256×32 (or 8192 bit) SRAM
Generated by hard-macro module generator

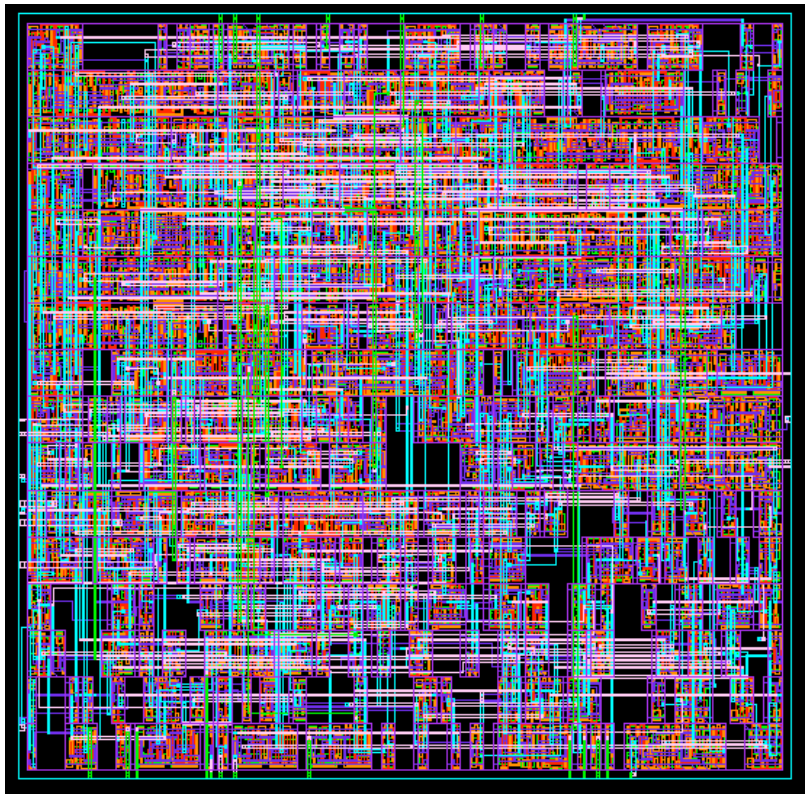
Dept. of Electrical and Computer Engineering
University of Connecticut

Macrocells

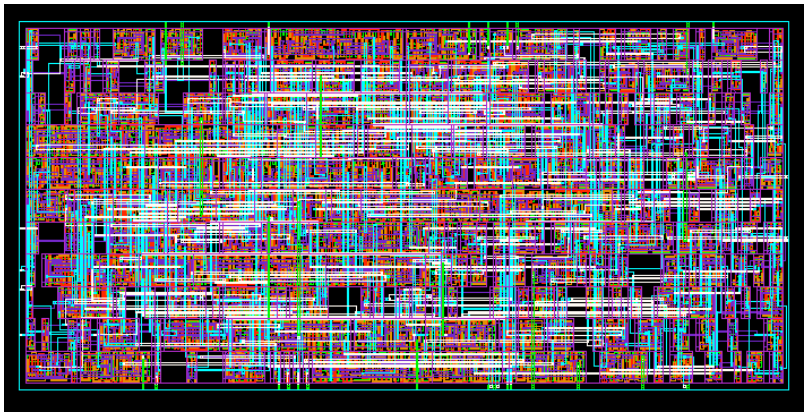
- Soft Macros
  - Physical design is done automatically
  - Easily ported across many different technologies and processes
  - Macro cell compiler will take a functional and parameterized description and generate a netlist of standard cells

Dept. of Electrical and Computer Engineering
University of Connecticut

Macrocells



```
string mat = "booth";
directive (multtype = mat);
output signed [16] Z = A * B;
```



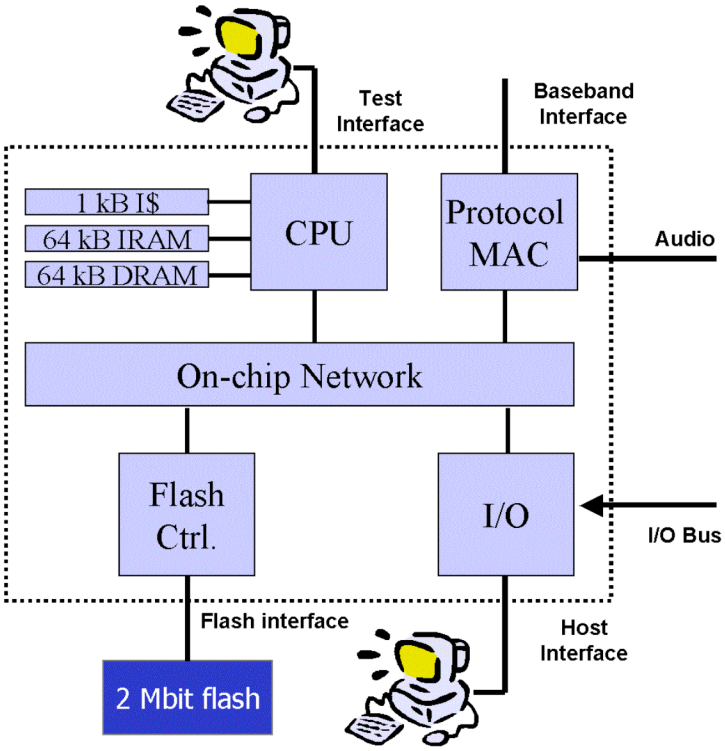
Dept. of Electrical and Computer Engineering
University of Connecticut

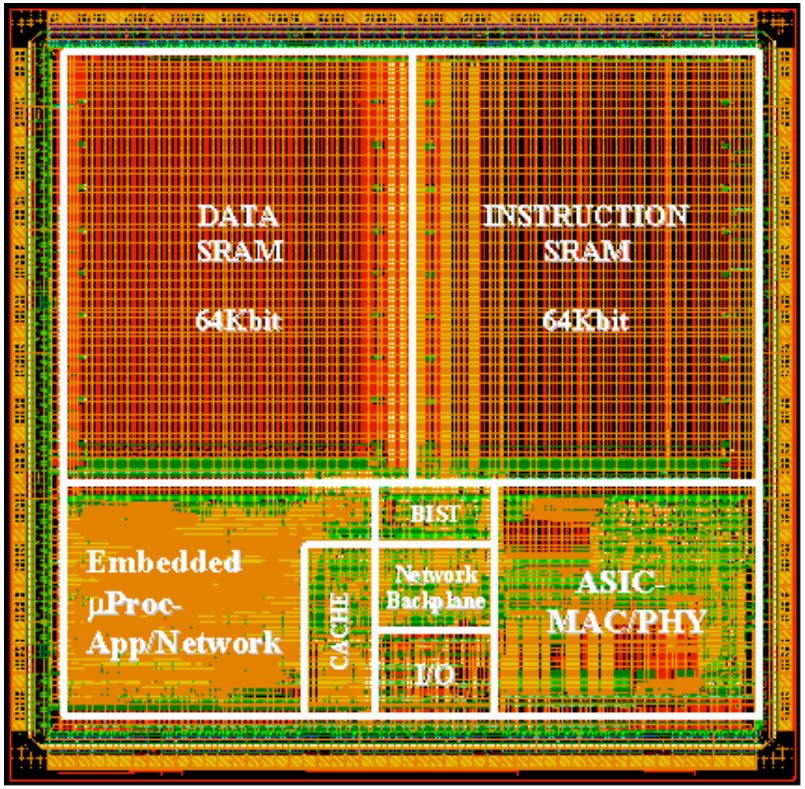
Intellectual property

- Macrocells can be aquired from third-party vendors
  - Includes appropriate compilers, debuggers, test vectors, prediction models
  - Similar to reusable software libraries
  - Examples include embedded processors, bus interfaces, DSP processors, ECC coders, MPEG codecs, etc.

Dept. of Electrical and Computer Engineering
University of Connecticut

“Intellectual Property”

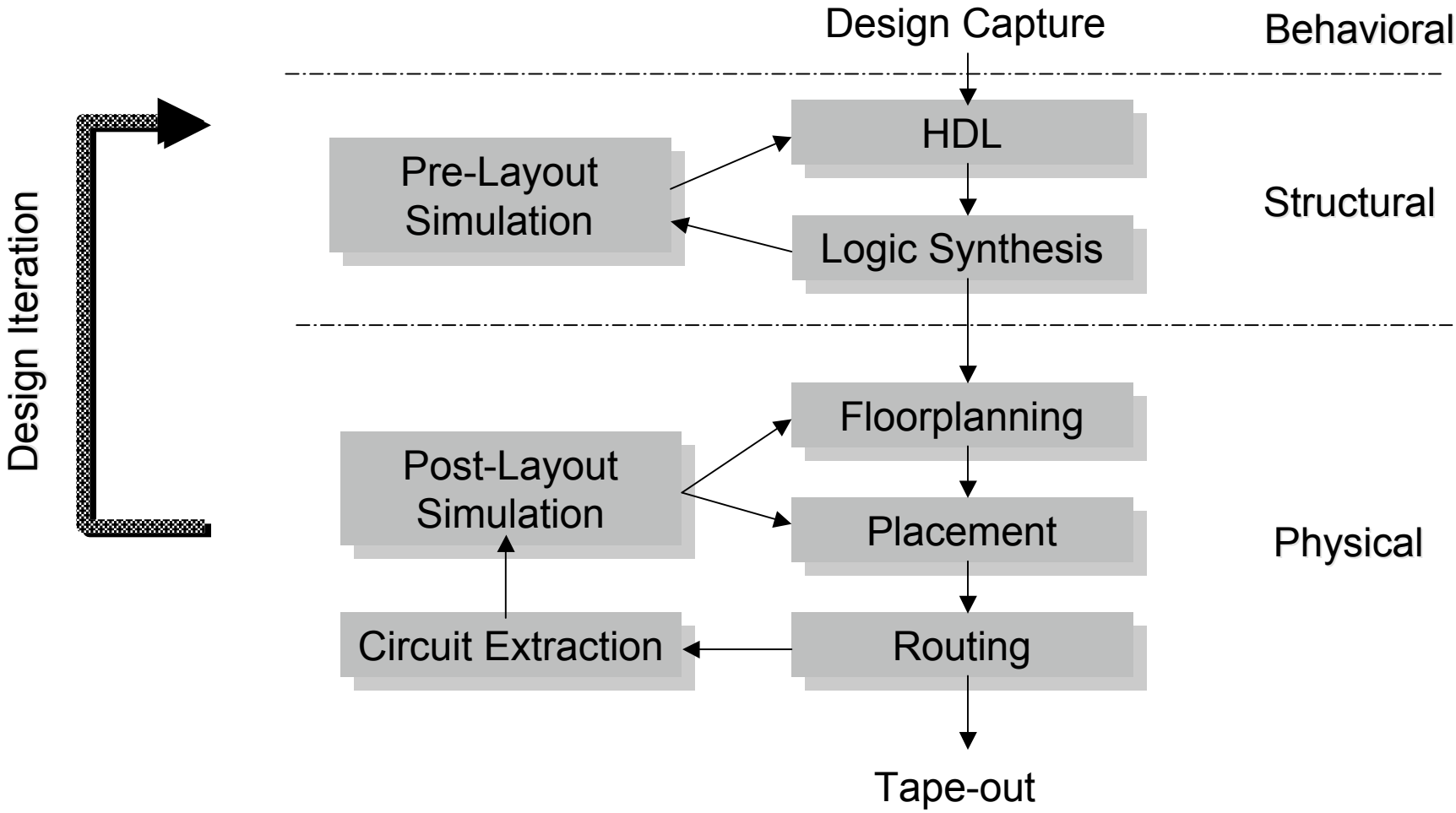




A Protocol Processor for Wireless

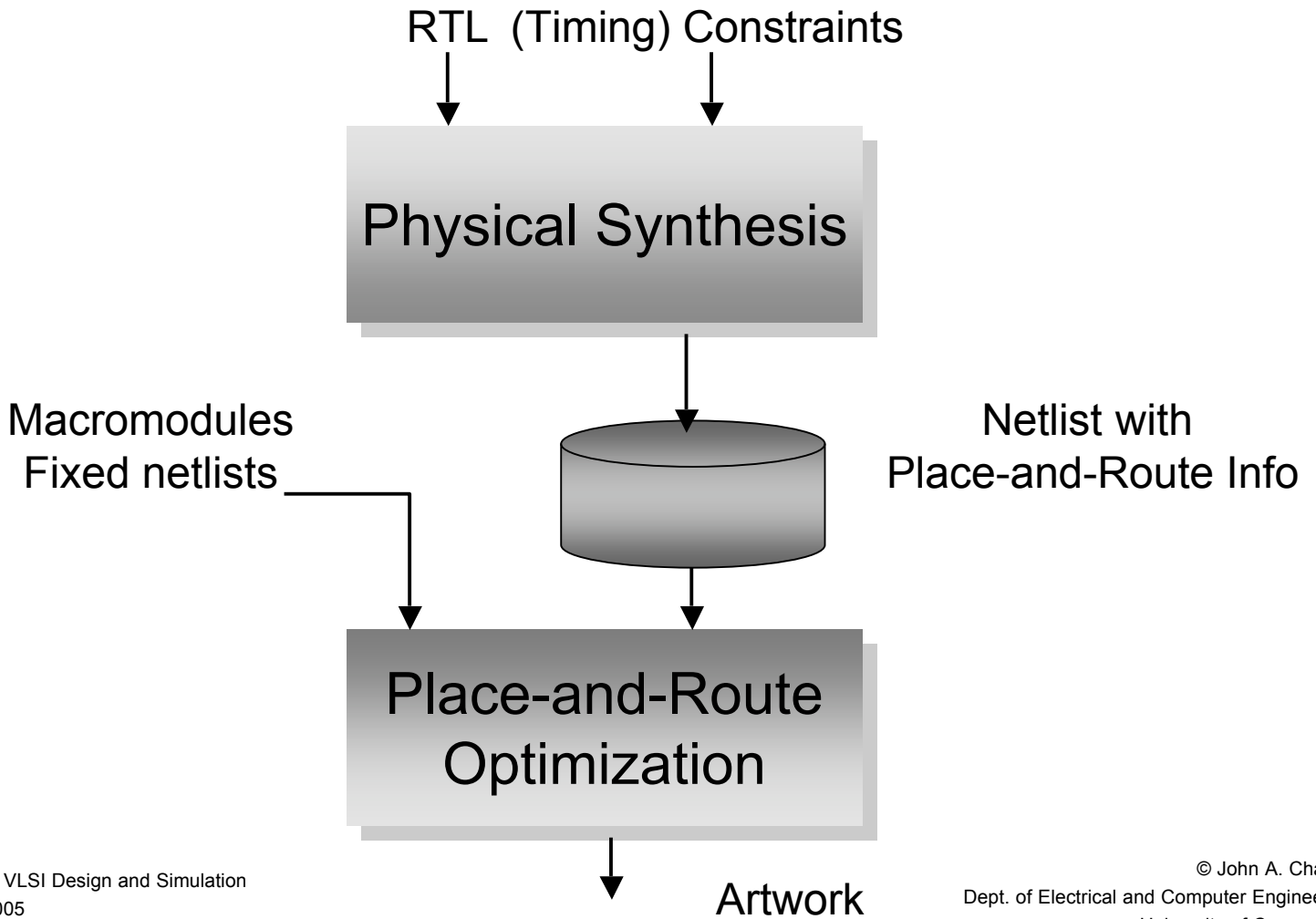
Dept. of Electrical and Computer Engineering
University of Connecticut

Semicustom Design Flow



Dept. of Electrical and Computer Engineering
University of Connecticut

Integrating Synthesis with Physical Design



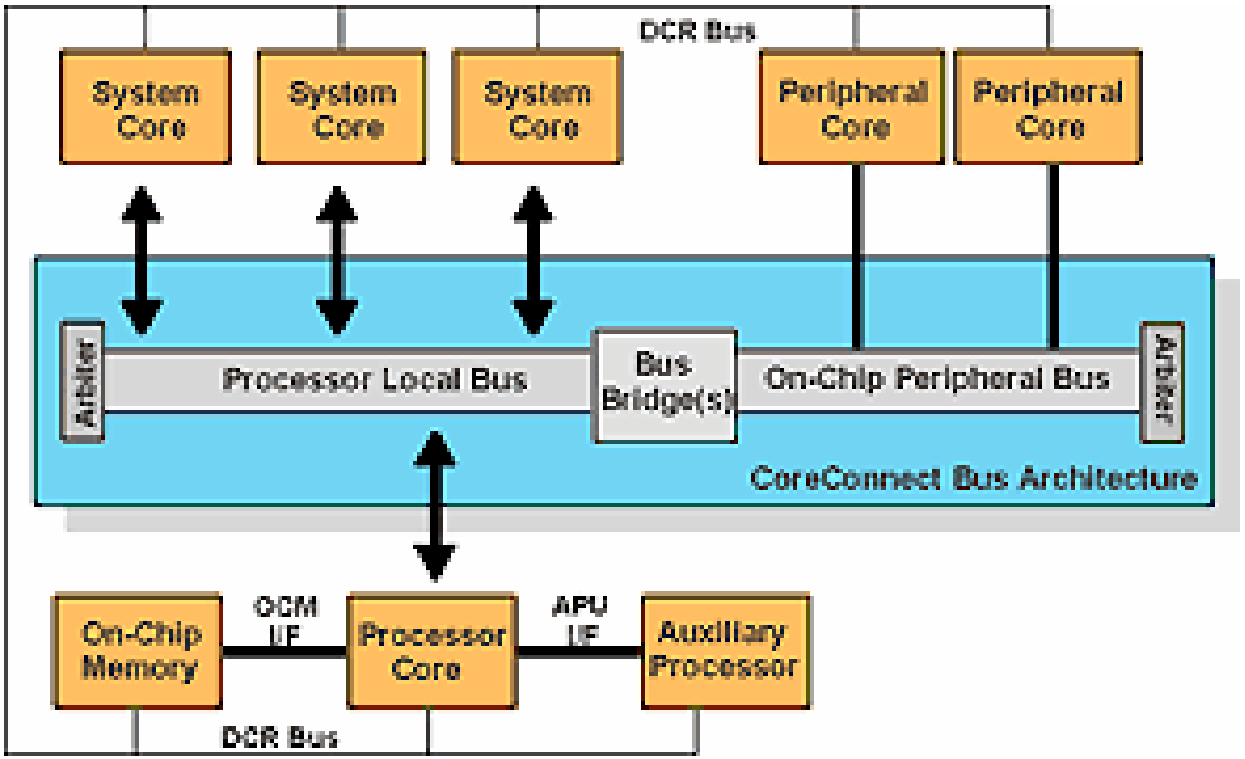
Dept. of Electrical and Computer Engineering
University of Connecticut

System-on-a-Chip (SoC) Design

- Embed multiple functionalities on a single chip
- SoC is a natural result of having more and more transistors available
- Managing multiple modules becomes a design challeng

Dept. of Electrical and Computer Engineering
University of Connecticut

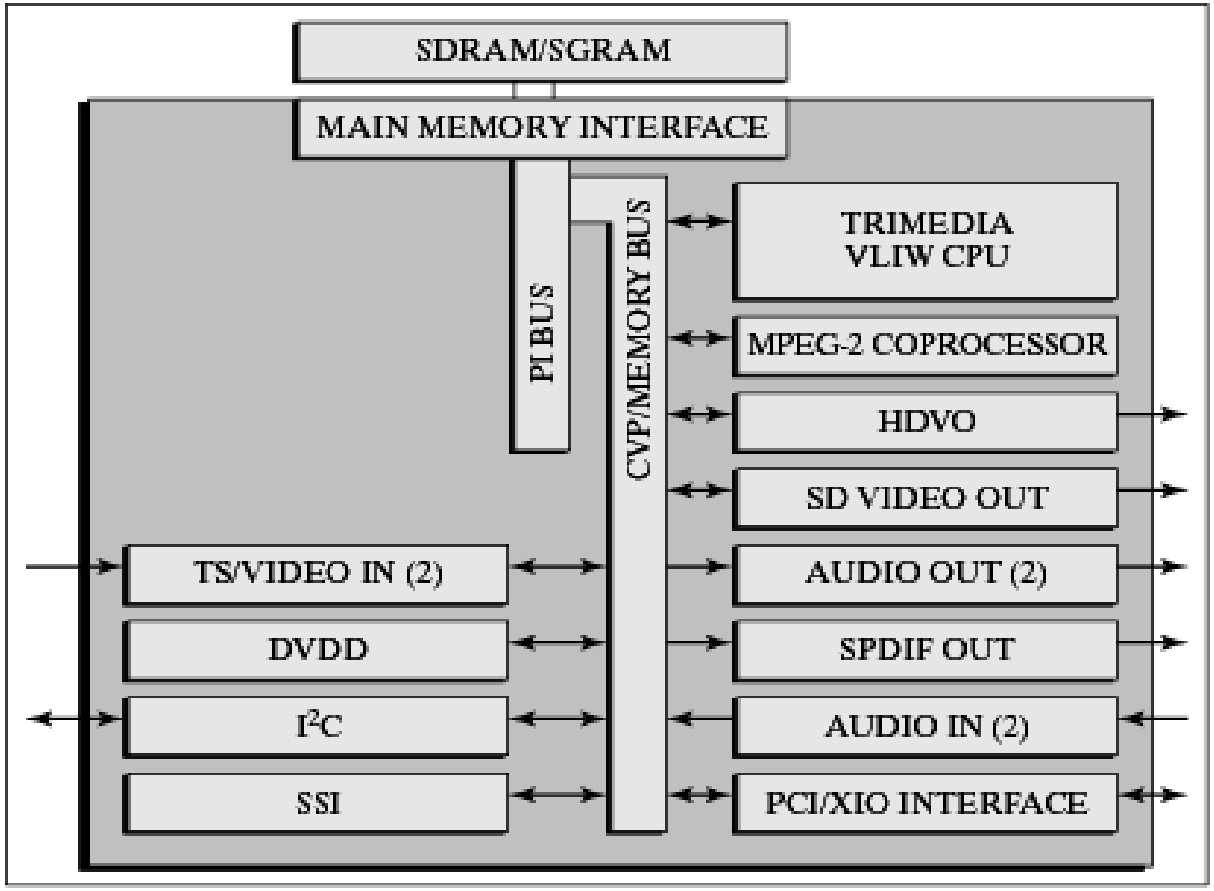
System-on-a-Chip Design



IBM CoreConnect Architecture

Dept. of Electrical and Computer Engineering
University of Connecticut

System-on-a-Chip Design



Philips Nexperia
HDTV SoC

Dept. of Electrical and Computer Engineering
University of Connecticut

System-on-a-Chip Design

- Embedded applications
- Mixed-mode applications (Analog/Digital)
- Heavy software component
- May have programmable and application specific components

Dept. of Electrical and Computer Engineering
University of Connecticut

Array-Based Design

- Cell-based and fully custom designs require a run through the full manufacturing process
  - Can take up to several months before the first chip arrives
  - Mask generation costs can make it very expensive
    - As process technologies get better, the tendency has been to use more and more masks
- Alternative is array-based implementations

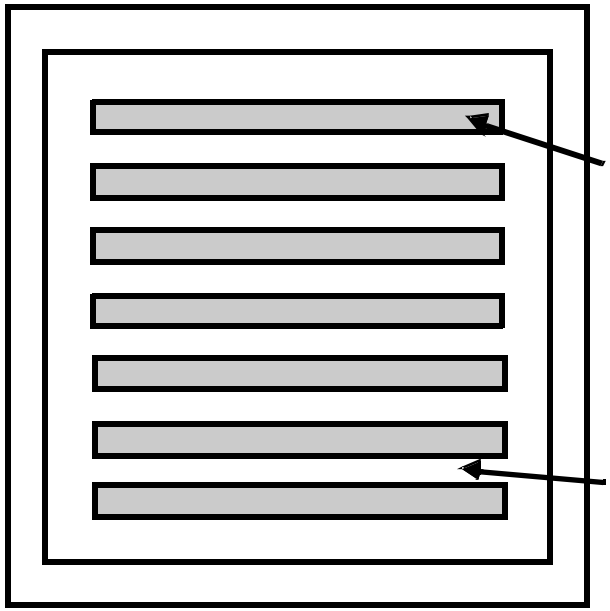
Dept. of Electrical and Computer Engineering
University of Connecticut

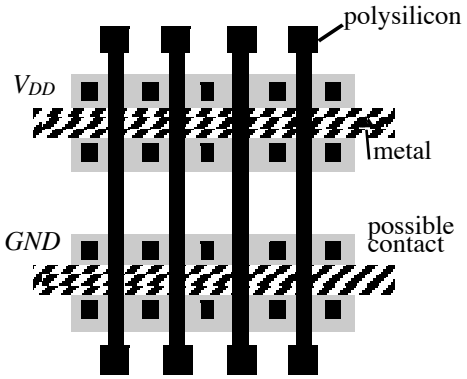
Array-Based Designs

- Pre-diffused
  - Mask-programmable
  - Gate arrays
  - Sea of gates
- Pre-wired
  - Field programmable gate arrays (FPGA)

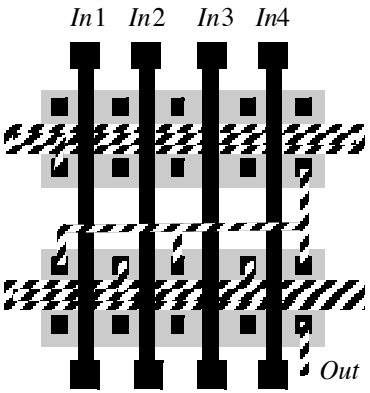
Dept. of Electrical and Computer Engineering
University of Connecticut

Gate arrays





Uncommited Cell



Committed Cell (4-input NOR)

Dept. of Electrical and Computer Engineering
University of Connecticut

Gate arrays

- Less compact than standard cells
- Manufacturing time savings is not as significant because the design times are the most important factor now
- Availability of FPGAs

Dept. of Electrical and Computer Engineering
University of Connecticut

Field Programmable Gate Arrays

- Prewired arrays
- Programming techniques
  - Write-once or fuse-based
  - Memory-based
    - Non-volatile (Flash, EEPROM, etc.)
    - Volatile (RAM)

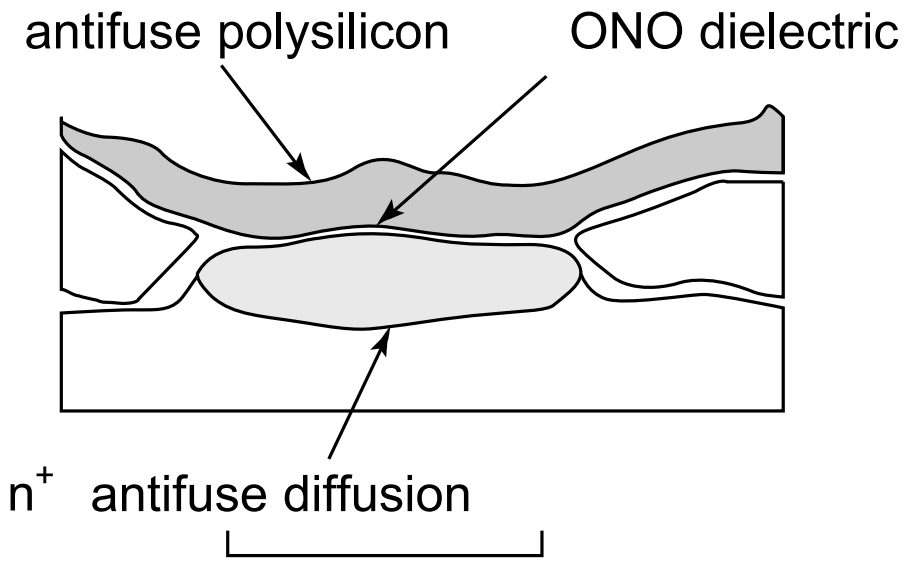
Dept. of Electrical and Computer Engineering
University of Connecticut

Field Programmable Gate Arrays

- Programmable Logic Style
  - Array-based
  - Cell-based
    - Function generator
    - Look-up table
- Programming Interconnect
  - Channel-routing
  - Mesh networks

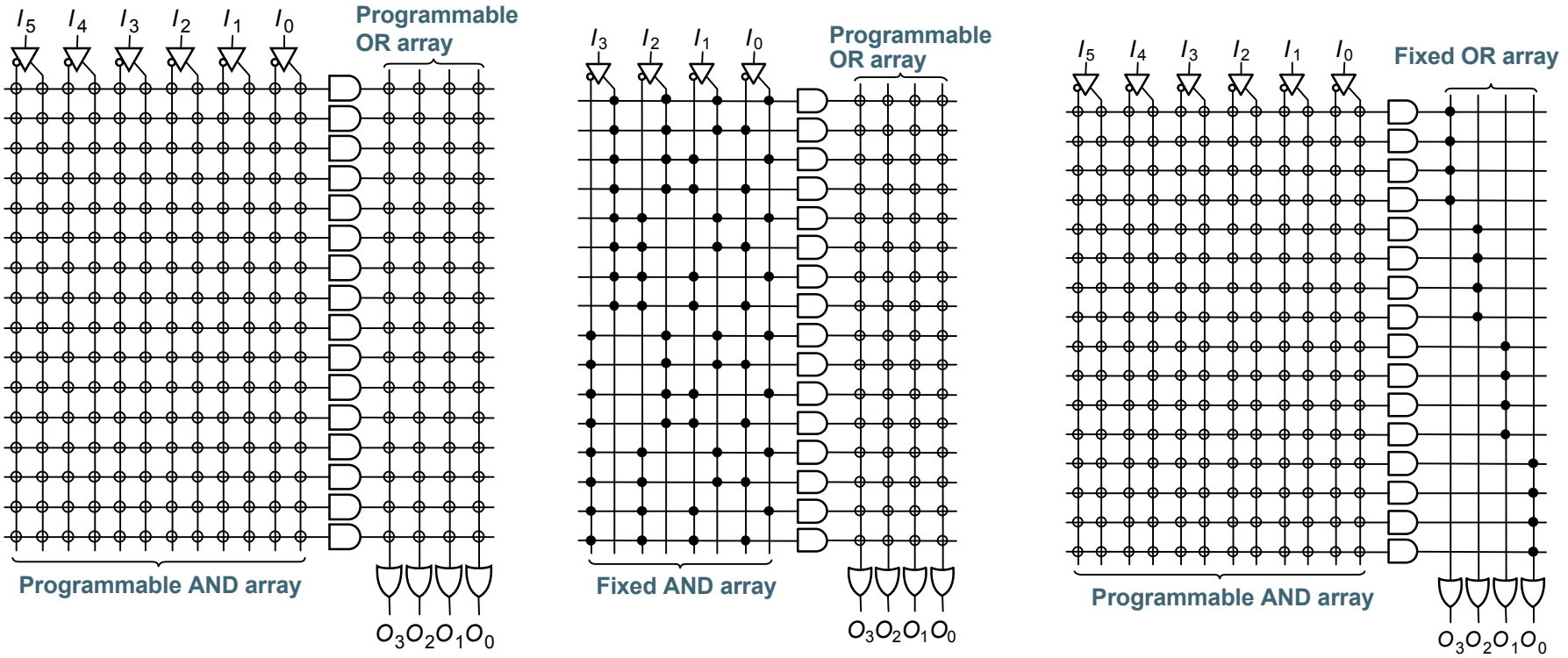
Dept. of Electrical and Computer Engineering
University of Connecticut

Fuse-based FPGA



Dept. of Electrical and Computer Engineering
University of Connecticut

Array-Based Programmable Logic



PLA PROM PAL

Indicates programmable connection

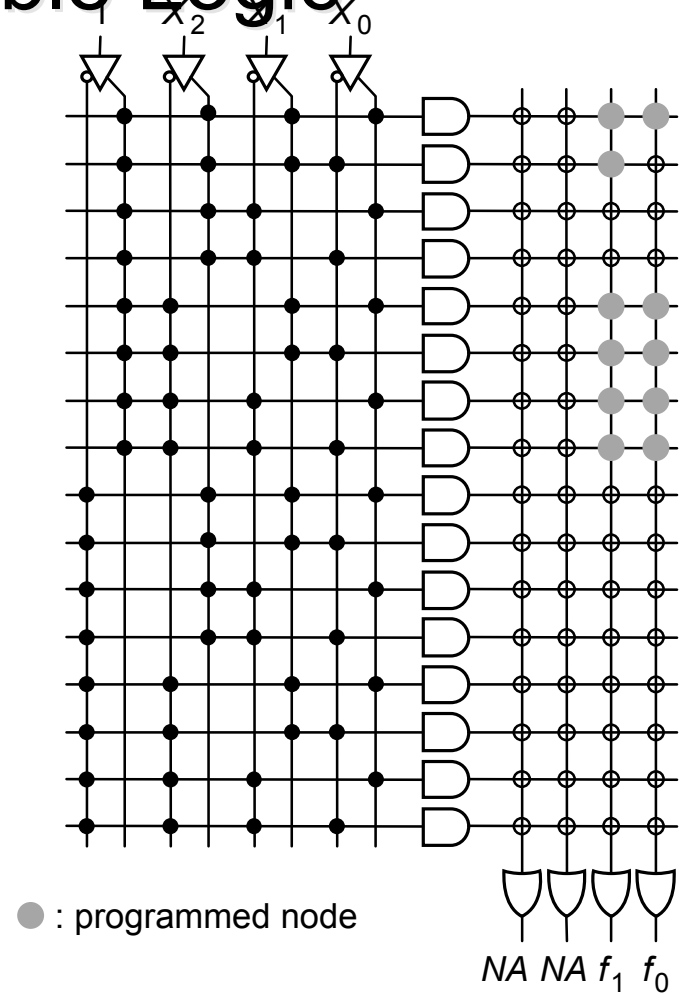
Indicates fixed connection

Dept. of Electrical and Computer Engineering
University of Connecticut

Array-Based Programmable Logic

$$f_0 = x_0 x_1 + \overline{x_2}$$

$$f_1 = x_0 x_1 x_2 + \overline{x_2} + \overline{x_0} x_1$$



Dept. of Electrical and Computer Engineering
University of Connecticut

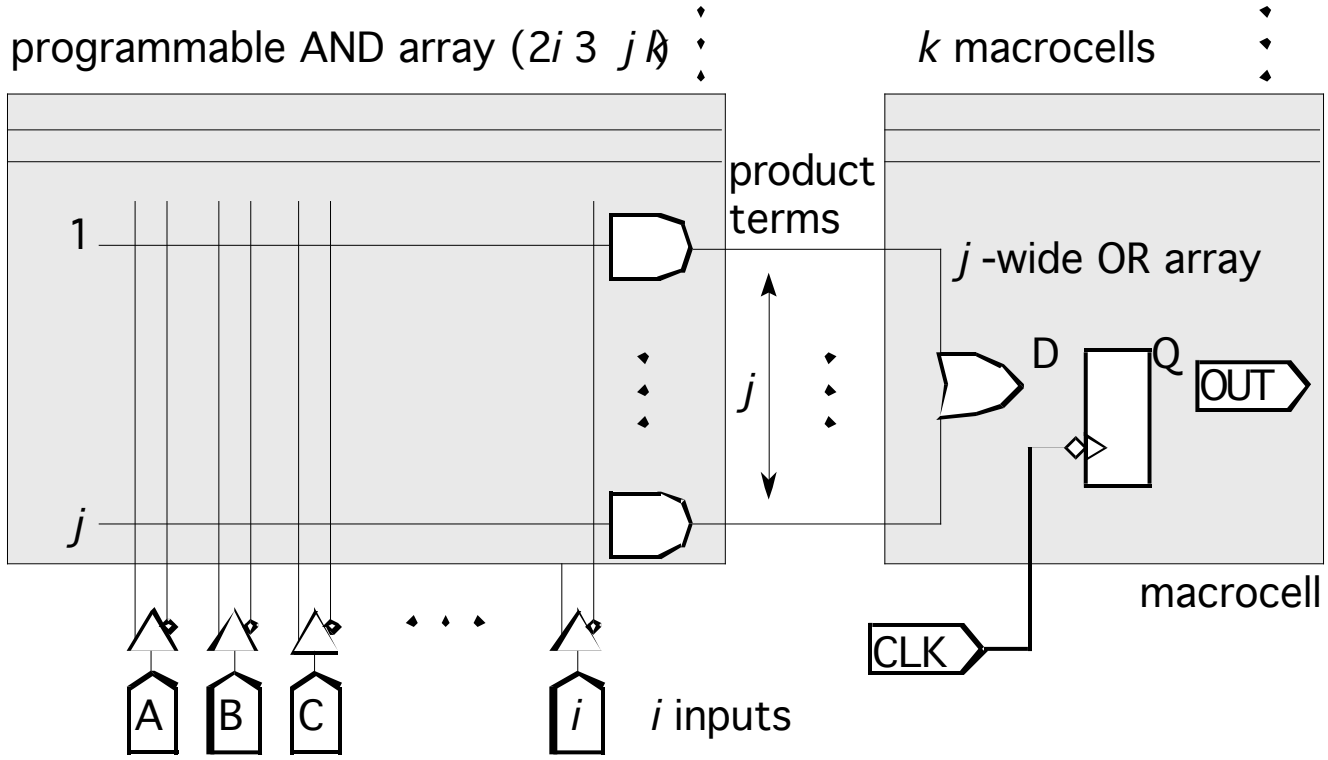
Array-Based Programmable Logic

- Lower density than custom
  - All minterms may not be actively used
  - OK if you have large fanin
- Lower performance
  - Each node has significant capacitance
- Only implements combinational logic - no registers or flip-flops

Dept. of Electrical and Computer Engineering
University of Connecticut

Array-Based Programmable Logic

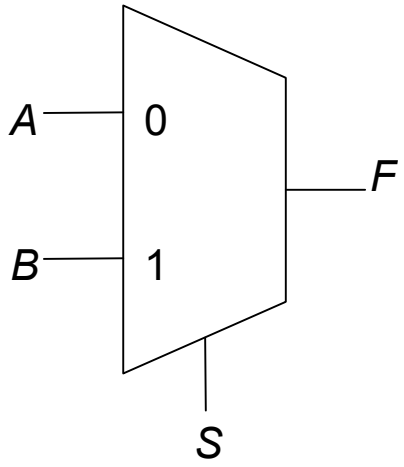
- Macrocell-based PAL



Dept. of Electrical and Computer Engineering
University of Connecticut

Cell-based Programmable logic

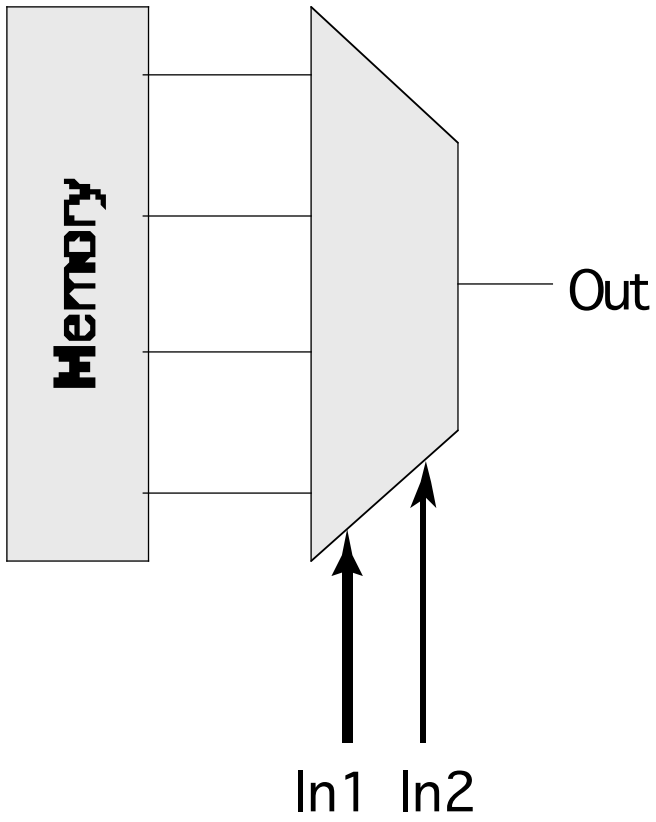
- Multiplexor as a function generator



Configuration			
A	***B***	***S***	***F=***
0	0	0	0
0	*X*	1	*X*
0	*Y*	1	*Y*
0	*Y*	*X*	*XY*
X	0	*Y*	$X\overline{Y}$
Y	0	*X*	$\overline{X}Y$
Y	1	*X*	$X + Y$
1	0	*X*	$\overline{X}$
1	0	*Y*	$\overline{Y}$
1	1	1	1

Dept. of Electrical and Computer Engineering
University of Connecticut

Lookup Table

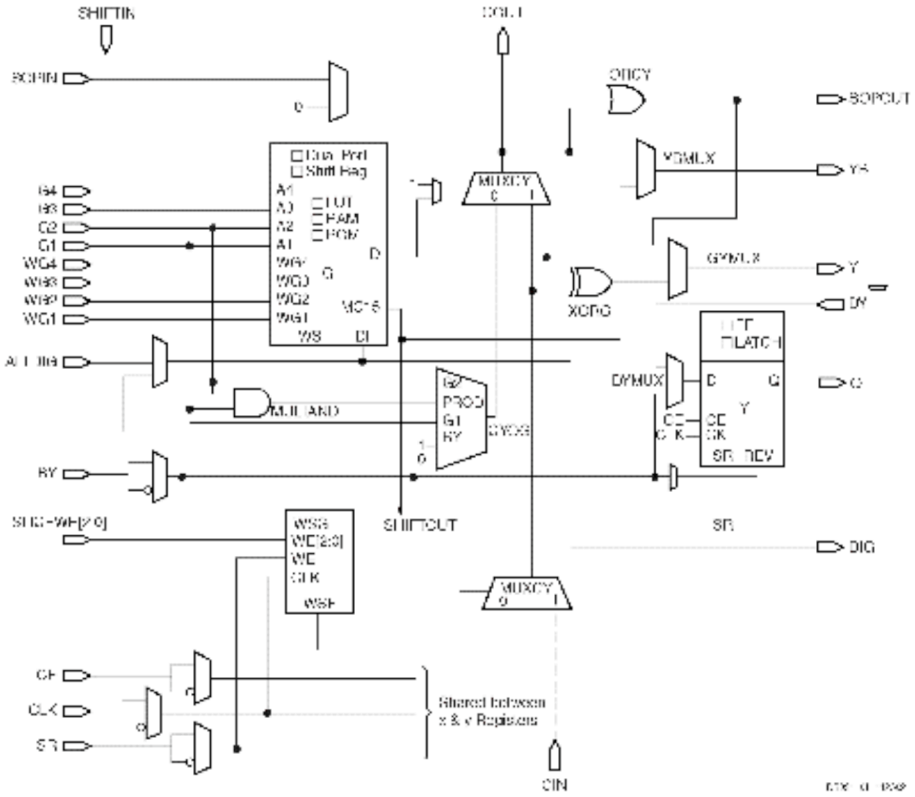


In	Out
00	0
01	1
10	1
11	0

Dept. of Electrical and Computer Engineering
University of Connecticut

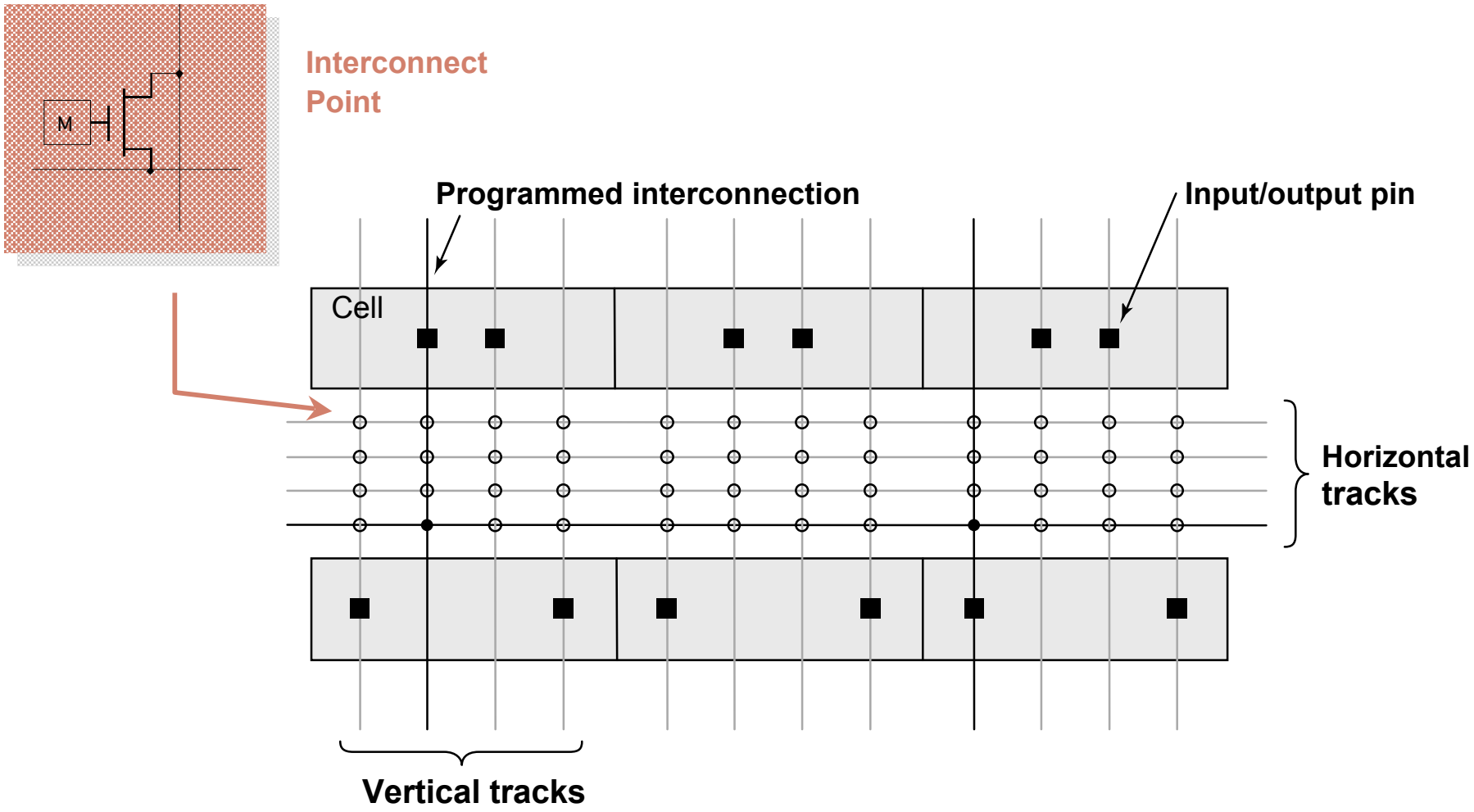
Lookup Table

- Configurable Logic Block (CLB) - Xilinx Virtex II



Dept. of Electrical and Computer Engineering
University of Connecticut

Array-based Programmable Wiring



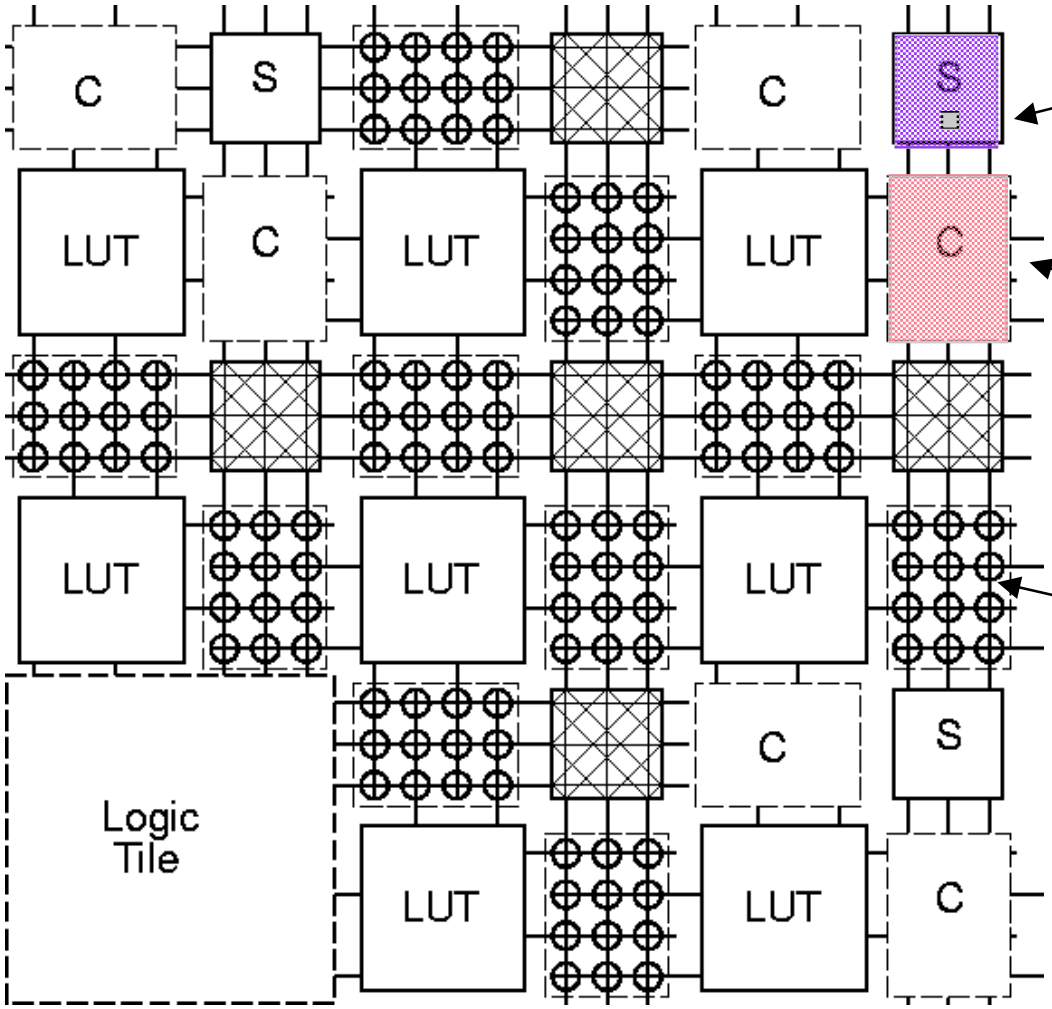
Dept. of Electrical and Computer Engineering
University of Connecticut

Array-based Programmable wiring

- Transistor method takes more space
- Use Fuse or antifuse methods
- Write-once - can not change

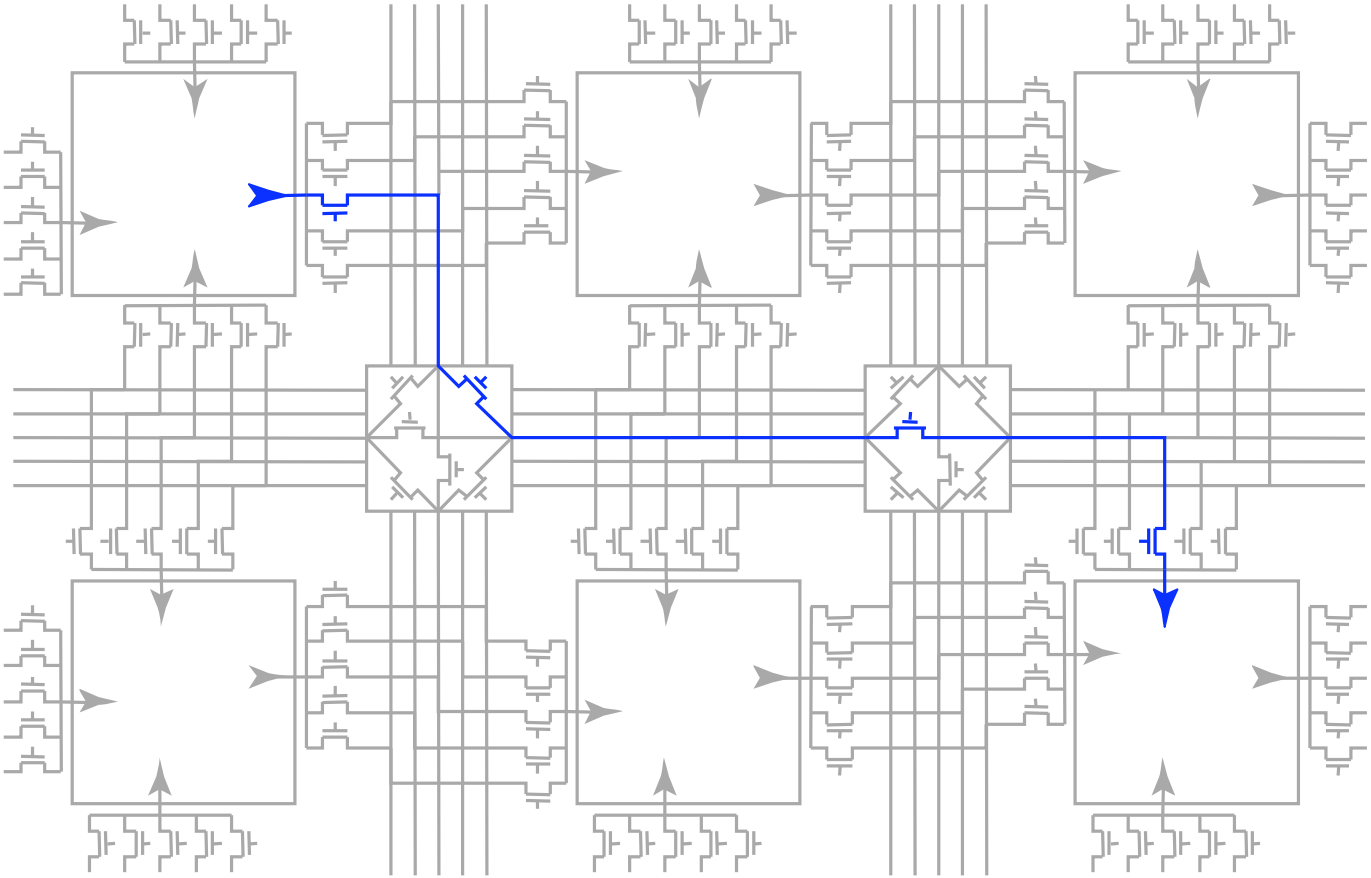
Dept. of Electrical and Computer Engineering
University of Connecticut

Mesh Network



Dept. of Electrical and Computer Engineering
University of Connecticut

Mesh Network



Dept. of Electrical and Computer Engineering
University of Connecticut

Mesh Network

- Sometimes can not route interconnect
- Pass transistor has a voltage drop
- RAM based progammable switch matrix
- Good for local wiring
- Global wiring will have large capacitive loads

Dept. of Electrical and Computer Engineering
University of Connecticut

Next class

- Testing and verification
- Exam 2 next Thursday

Dept. of Electrical and Computer Engineering
University of Connecticut