

# CSCE 611

## Digital Systems Design I

---

2005/3/27

### Week 12a Supplemental Notes

### Architecture Design of Regular Structures

© 2004 Dr. James P. Davis

Some figures: Ashenden © 2003 Morgan Kaufmann Publishers, Inc.  
Lewin & Protheroe © 1992 Chapman Hall Publishers, Ltd.

---

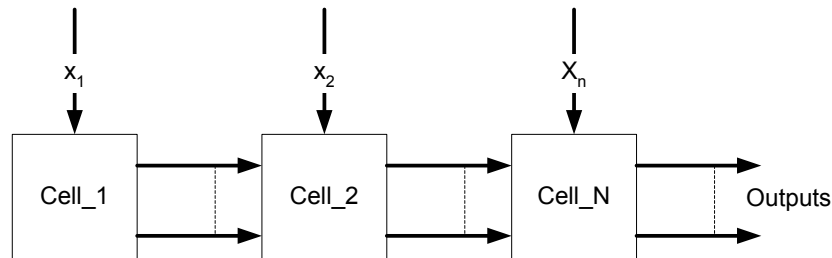
## Outline

- Key points:
  - ✓ Design and use of “regular” structures that reuse a design element, connected together iteratively or recursively, realizing some functionality in the digital system.
  - ✓ We can reduce complexity and cost of our designs by identifying and constructing subsystems, which can then be repeated or replicated to form the complete system.
  - ✓ The regular repeating (or iterative) nature of circuits refers to their structure, consisting of a number of identical components (sometimes called “cells”) that are linked to form an “array” that carries out the overall function.
  - ✓ Look at 2 different examples using ASM modeling:
    - ◇ Memory array (memory modeling in Nimbus).
    - ◇ Multiplier circuits – we’ll compare the architecture of this multiplier model with that of the shift-add MUL model.

# Iterative Array Structure

---

- Iterative structures are repeating components that are identical components linked to perform the overall function.
- (+) using this type of architecture reduces design time.
- (-) using this style increases propagation delay through all stages of the array (which can be mitigated by using pipelining).



Source: Protheroe & Lewin © 1992, Chapman Hall



© 2004 Dr. James P. Davis Page 3

# Iterative Array Structure

---

- Characteristics of iterative arrays:
  - ✓ Technique is most often used in applications where the overall function is carried out by performing the same operation on a sequence of data bits (e.g., SIMD architecture pattern).
  - ✓ This pattern is well-suited to pattern recognition and encoding and decoding circuits with a large number of parallel inputs.
  - ✓ Circuits containing any number of variables can be constructed by extending the number of regular cells in the circuit.
  - ✓ Pipelining stages of the iterative array: use a clock to step successive values through each array cell.
  - ✓ Any n-stage sequential circuit can be realized an n-stage iterative array; each stage consists of a copy of the combinational circuit implementing the next-state and output equations of the sequential circuit.

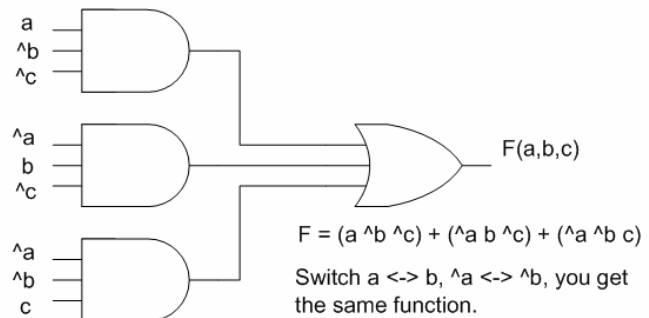


© 2004 Dr. James P. Davis Page 4

# Symmetric Circuit Structure

Source: Protheroe & Lewin © 1992, Chapman Hall

- Definition: a symmetric logic function has the property of remaining unaltered when any two of its variables are interchanged.
- Symmetric functions with un-complemented variables of symmetry are called *n-out-of-m functions*: i.e., the logic function generates a '1' output when exactly *n* of its *m* inputs are detected as being '1'.



© 2004 Dr. James P. Davis Page 5

# Cellular Array Structure

- Definition:
  - ✓ A cellular array structure employs a replicated component that carried out a computation, which is interconnected to other "cells" by "rails" or a "mesh".
  - ✓ One-dimensional array uses single or multiple rails to connect the cells in the array topology.
  - ✓ Two-dimensional array uses a mesh interconnect structure to connect the cells to form a computational "fabric".
- Classification:
  - ✓ Type of cell: whether it performs the same or different functions.
  - ✓ Dimension: one or two dimensional arrays are common.
  - ✓ Data flow: unilateral (one direction) or bilateral (both directions simultaneously).
- This structure is modular, and yields simple and regular circuits that support parallel computation.

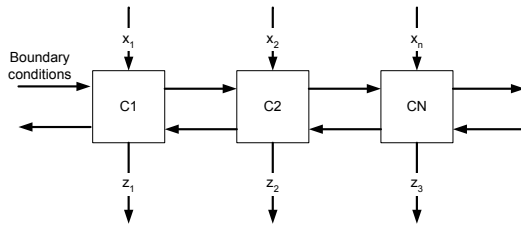
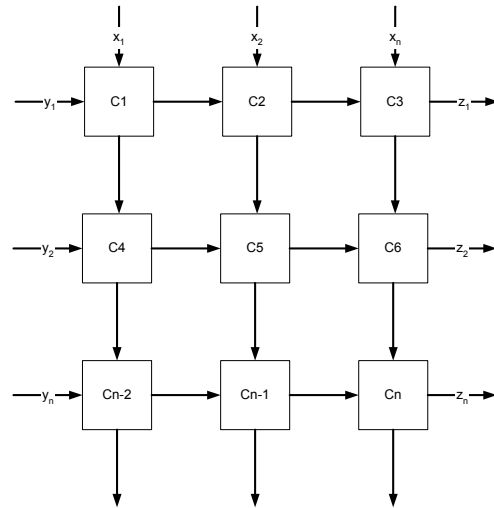


© 2004 Dr. James P. Davis Page 6

# Cellular Array Structure

Source: Protheroe & Lewin © 1992, Chapman Hall

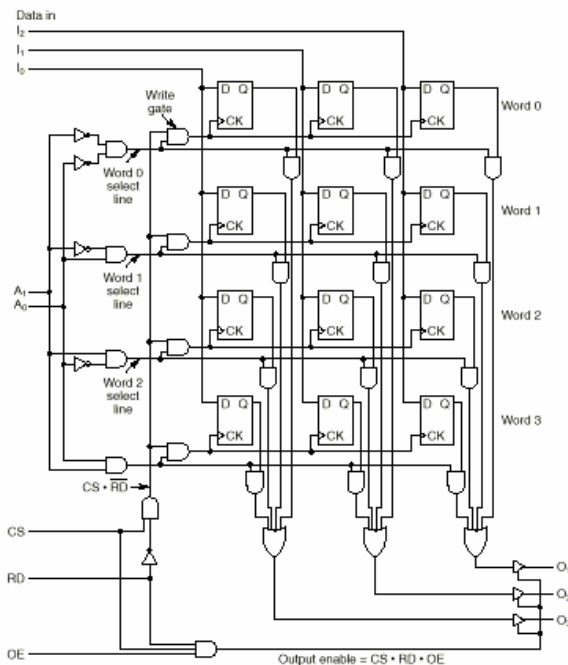
- Cellular array structures are also referred to as *systolic* arrays.
  - One-dimensional bilateral array.
  - Two-dimensional unilateral array.
- Specify:
  - Primary inputs & outputs
  - Boundary conditions (unspecified entering edges of array).



© 2004 Dr. James P. Davis Page 7

# Regular Array – Memory Example

Source: Tanenbaum, 4<sup>th</sup> ed. © 1999, Prentice-Hall



- 4 x 3 Memory Array
  - The Memory array is built up from gates and flip flops, to take advantage of certain properties of the devices. Each row is one of four 3-bit words.
  - Data Lines  $I_0 - I_2$  feed all of the D FF's in a column. The address lines  $A_0 - A_1$  act as select lines for a given bank.
  - The control signals CS (control select), RD (read enable), OE (output enable) are used to route data to and from memory (allowing writes and reads) based on the combinational logic gates.
  - The bus "drivers" are enabled by the AND of the 3 control signals.

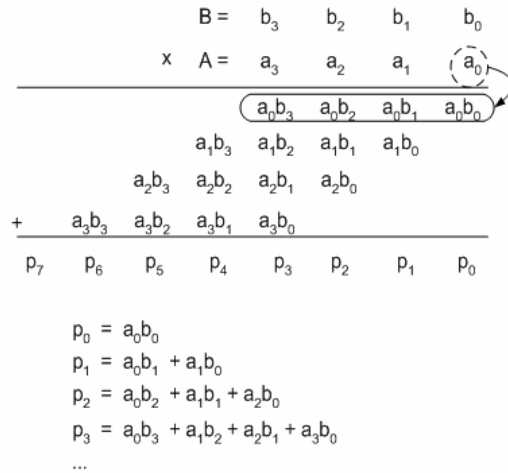


© 2004 Dr. James P. Davis Page 8

# Multiplication Example – pen and paper

Source: Protheroe & Lewin © 1992, Chapman Hall

- Shift-add multiplication:
  - ✓ Take each digit of *multiplicand* operand, and multiply it by the first digit of the *multiplier* operand.
  - ✓ Bring each *partial product* term down into its column.
  - ✓ Repeat, multiplying each digit of *multiplicand* with each subsequent digit of *multiplier* operand, bring the *partial product* terms down onto the column and row.
  - ✓ Add the *partial product* terms, column by column, to generate the *product*.

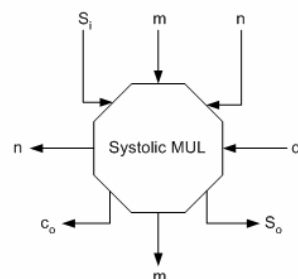
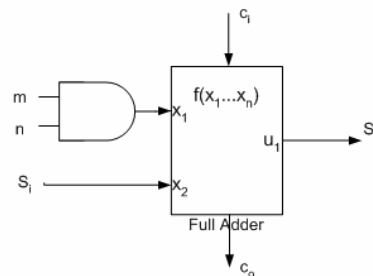


© 2004 Dr. James P. Davis Page 9

# Regular Array – Multiplier Example

Source: Protheroe & Lewin © 1992, Chapman Hall

- Multiplier structure:
  - ✓ **Cell structure:** consists of a *Full Adder*, with one operand input tied to AND gate, and 2<sup>nd</sup> operand available for connection to Sum  $S_i$  of a previous cell.
  - ✓ Inputs  $m, n$  are (1) the adder operands, and (2) the “gating” input from a cell that is tied to this one, indicating that it is linked in an array.
  - ✓ **Cell interface:** consists of inputs and outputs, as denoted on the internal model (above) with the addition of pass-through outputs for both  $m, n$ .

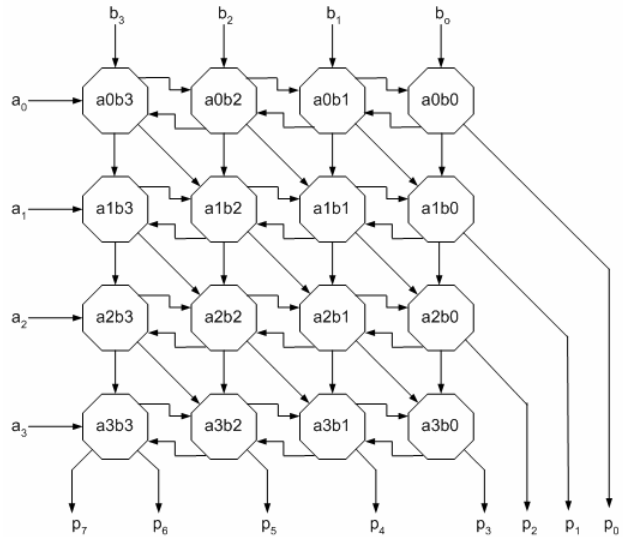


© 2004 Dr. James P. Davis Page 10

# Regular Array – Multiplier Example

Source: Protheroe & Lewin © 1992, Chapman Hall

- Multiplier array:
  - ✓  $m$  inputs: digit values of b operand (multiplier).
  - ✓  $n$  inputs: digit values of a operand (multiplicand).
  - ✓  $S_i$  inputs: for boundary cells, it is tied to zero, for inner cells, it is tied to  $S_0$  of preceding cell on the diagonal.
  - ✓  $S_0$  outputs: for inner cells, tied to  $S_i$  of succeeding cell along diagonal; for outer cells, it generates each of the recombinant product terms.
  - ✓  $C_i, C_o$  terms: this is the carry in and carry out terms, generated from the Full Adder operations on the inputs.



© 2004 Dr. James P. Davis Page 11

## Array-based Architecture Patterns

- Taxonomy of patterns:
  - ✓ Combinational logic array circuit structures
    - ✦ Iterative circuits (e.g., SIMD type of architecture).
    - ✦ Symmetric function circuits.
    - ✦ Cellular array circuits.
  - ✓ Sequential circuit structures
    - ✦ Can be “unrolled” into a sequence of iterative structures.
    - ✦ Pipeline arrays: use clocked registers in stages of computation. Successive input values are applied one each new clock cycle.
- Metrics:
  - ✓ Propagation delay of one cell:  $t_d$ , for the whole array:  $t_a$ .
  - ✓ Generating output of stage  $n$  of the pipeline:  $t_a = n * t_d$ .
  - ✓ Throughput of the pipeline array:  $1 / t_d$ .
  - ✓ Latency of the pipeline array:  $t_a$ .



© 2004 Dr. James P. Davis Page 12