

Lecture ~~16~~ 17  
Course Summary and Additional Topics

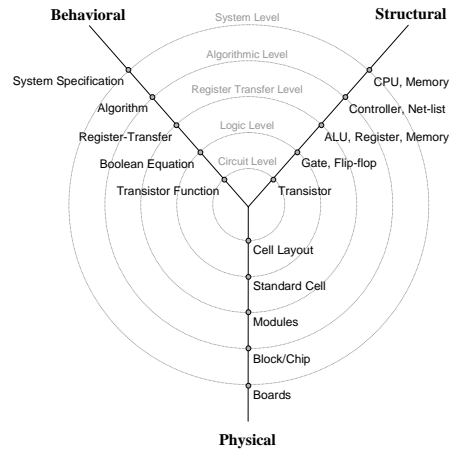
## What Have We Done?

---

- Design Specification
  - Structural vs. Behavioral
  - Design and implementation of sequential circuits – RTL (Register-Transfer Level) Design
  - C to gates
- Optimization Techniques
  - Optimization and tradeoffs of combinational and sequential circuits
  - Heuristic minimization of two-level circuits
  - Binary decision diagrams (BDDs)
  - *Multi-Level Minimization*
- CAD Tools
  - Utilize industry standard tools to simulate (and synthesize) design
  - Develop optimization tools

## Is That All There Is?

- Remember the Y-chart
  - Design tasks can be performed at different abstractions
- Topics we studied are no different
  - Design Specification
  - Optimization Techniques
  - CAD Tools
- We looked at quite a few, but much more exists

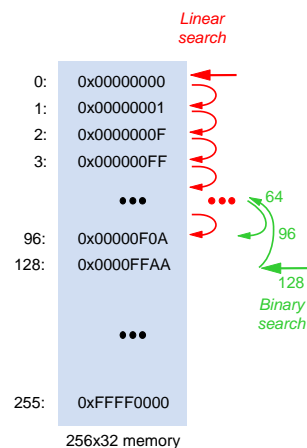


ECE 474a/575a  
Susan Lysecky

3 of 13

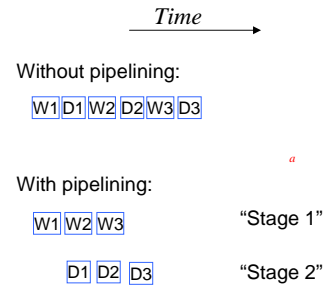
## Algorithm Selection

- Chosen algorithm can have big impact
- Example: Quickly find item's address in 256-word memory
  - Algorithm 1: "Linear search"
    - Compare item with  $M[0]$ , then  $M[1]$ ,  $M[2]$ , ...
    - 256 comparisons worst case
  - Algorithm 2: "Binary search" (sort memory first)
    - Start considering entire memory range
      - If  $M[\text{mid}] > \text{item}$ , consider lower half of  $M$
      - If  $M[\text{mid}] < \text{item}$ , consider upper half of  $M$
      - Repeat on new smaller range
      - Dividing range by 2 each step; at most 8 such divisions
    - Only 8 comparisons in worst case
- Choice of algorithm has *tremendous* impact
  - Far more impact than say choice of comparator type
  - Research looks at how to write algorithm intended to run of HW vs. SW



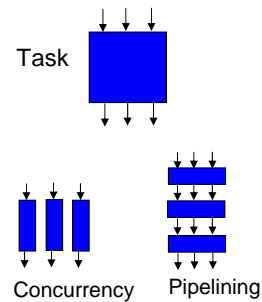
# Pipelining

- Intuitive example: Washing dishes with a friend, you wash, friend dries
  - You wash plate 1
  - Then friend dries plate 1, *while you wash plate 2*
  - Then friend dries plate 2, while you wash plate 3; and so on
  - You don't sit and watch friend dry; you start on the next plate
- Pipelining:** Break task into stages, each stage outputs data for next stage, all stages operate concurrently (if they have data)

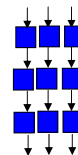


# Concurrency

- Concurrency:** Divide task into subparts, execute subparts simultaneously
  - Dishwashing example: Divide stack into 3 substacks, give substacks to 3 neighbors, who work simultaneously -- 3 times speedup (ignoring time to move dishes to neighbors' homes)
  - Concurrency does things side-by-side; pipelining instead uses stages (like a factory line)

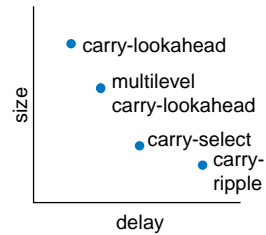


Can do both, too

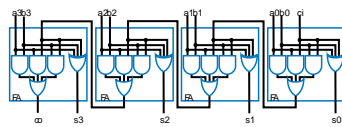


## Component Level Optimization and Tradeoffs

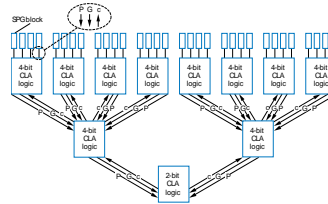
- Designer picks the adder that satisfies particular delay and size requirements
  - May use different adder types in different parts of same design
  - Faster adders on critical path, smaller adders on non-critical path



carry-ripple adder

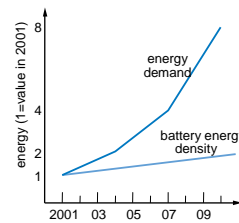


carry-lookahead adder



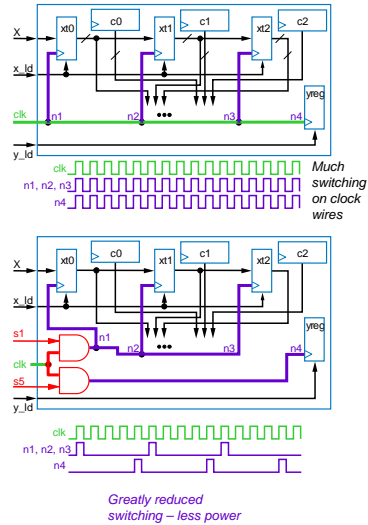
## Power Optimization

- Until now, we've focused on size and delay
- **Power** is another important design criteria
  - Measured in Watts (energy/second)
    - Rate at which energy is consumed
- Increasingly important as more transistors fit on a chip
  - Power not scaling down at same rate as size
    - Means more heat per unit area – cooling is difficult
    - Coupled with battery's not improving at same rate
      - Means battery can't supply chip's power for as long
  - CMOS technology: Switching a wire from 0 to 1 consumes power (known as **dynamic power**)
    - $P = k * CV^2f$ 
      - k: constant; C: capacitance of wires; V: voltage; f: switching frequency
  - Power reduction methods
    - Reduce voltage: But slower, and there's a limit
    - What else?



## Power Optimization using Clock Gating

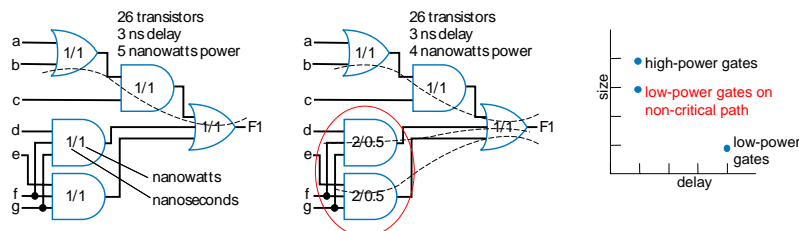
- $P = k * CV^2f$
- Much of a chip's switching  $f$  (>30%) due to clock signals
  - After all, clock goes to every register
  - Portion of FIR filter shown on right
    - Notice clock signals n1, n2, n3, n4
- Solution: Disable clock switching to registers unused in a particular state
  - Achieve using AND gates
  - FSM only sets 2<sup>nd</sup> input to AND gate to 1 in those states during which register gets loaded
- Note: Advanced method, usually done by tools, not designers
  - Putting gates on clock wires creates variations in clock signal (*clock skew*); must be done with great care



## Power Optimization

### Low-Power Gates on Non-Critical Paths

- Another method: Use low-power gates
  - Multiple versions of gates may exist
    - Fast/high-power, and slow/low-power, versions
  - Use **slow/low-power gates on non-critical paths**
    - Reduces power, without increasing delay

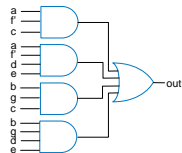


# Multi-level Logic Optimization

## Factoring

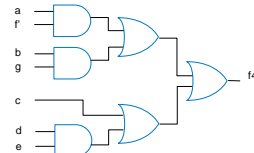
- Logic equation in a form with more than two levels of logic
  - Many instances when smaller input gates require less area and power

12a.  $x \cdot (y + z) = xy + xz$   
 12b.  $x + y \cdot z = (x+y) \cdot (x+z)$

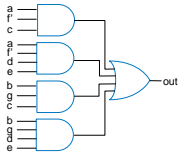


out =  $af'c + af'de + bgc + bgde$

$f = af'c + af'de + bgc + bgde$   
 $f = af'(c + de) + bg(c + de)$   
 $f = (af' + bg)(c + de)$

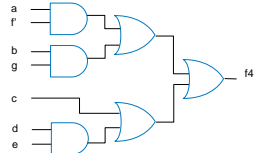


$f = (af' + bg)(c + de)$



out =  $af'c + af'de + bgc + bgde$

$f = af'c + af'de + bgc + bgde$   
 $f = af'(c + de) + bg(c + de)$   
 $f = (af' + bg)(c + de)$



$f = (af' + bg)(c + de)$

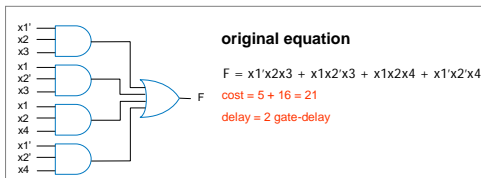
ECE 474a/575a  
Susan Lysecky

11 of 13

# Multi-level Logic Optimization

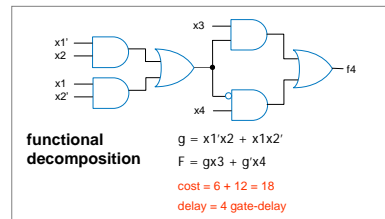
## Functional Decomposition

- Functional Decomposition
  - Replace two-level logic equation with two or more new expressions which are then combined to define multilevel circuit



**original equation**

$F = x1'x2x3 + x1x2'x3 + x1x2x4 + x1'x2'x4$   
 cost =  $5 + 16 = 21$   
 delay = 2 gate-delay



**functional decomposition**

$g = x1'x2 + x1x2'$   
 $F = gx3 + g'x4$   
 cost =  $6 + 12 = 18$   
 delay = 4 gate-delay

$F = x1'x2x3 + x1x2'x3 + x1x2x4 + x1'x2'x4$   
 $F = (x1'x2 + x1x2')x3 + (x1x2 + x1'x2')x4$   
 $g' = x1x2 + x1'x2'$   
 $F = gx3 + g'x4$

Let  $g(x1, x2) = x1'x2 + x1x2'$

Observe  $g'$

Re-write the equation using  $g$



ECE 474a/575a  
Susan Lysecky

12 of 13

# There is much more to CAD

- Overview of synthesis and optimization of digital circuits
  - Process by which a higher-level design specification is converted to a lower-level specification
  - Several levels of synthesis within digital design

