Matthew Tran
EE151
MT1
4 up 1 before REFOR

- Moore's Law - 2x transistor/ 27 mo, scaling
- Dennard Scaling - how scale thru w/ transistor size
- Frequency flat, power flat
- Pareto Optimal Frontier - best can do at any point
- performance vs cost vs speed tradeoff
- abstraction, modular design
  - top down vs bottom up
- thermodynamics - energy/time
- latency - time
- energy, power
- NRE cost (one time) vs Recurring costs
- Die(chip) cost = $\frac{\text{wafer cost}}{\text{die/wafer} \cdot \text{die yield}}$

$$\text{die/wafer} = \frac{\pi \cdot (r_{wafer})^2}{\text{die area}} - \frac{\pi \cdot d_{wafer}}{\sqrt{2 \cdot \text{die area}}}$$

$$\text{die yield} = \left(1 + \frac{\text{defect/unit area} \cdot \text{die area}}{\alpha}\right)^{-\alpha}$$

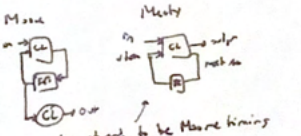
$\alpha \approx 3$

$$\text{die cost} \propto F(\text{die area})^4$$

- abstraction/separation - "clean up" signals
  - relieve nondeterminacy
- combinational logic (CL) - just inputs
- sequential logic - some of output/prev state for
  - usually on clock tick
- state element - stores info
- RTL - abstraction CL + state elements
- Implementation
  - Full custom
  - Standard-cell (usually ASIC)
  - Gate-array
  - FPGA
  - Microprocessor
  - Domain specific processor
- "hardware thread" - extends when functionality?
- Hierarchy shown
- Verilog
  - an HDL
  - structural - actual wires, one-to-one
  - behavioral - high level connections
    - tuples, arrays
  - module <name> (<port list>);
    <port def>
    end module
    use named ports
    <name> <instance name> (.<conn>);
  - reduction op - on all bits {{ }} concat
  - unsigned by default {{{ }}} repeat
    - add signed if want
  - continuous assignment - assign = ... ;
  - non-continuous assign
    - always @( ) begin (sensitivity list)
      ... = ...
      ↑ reg type (not necessarily wire)
      end
    - allows case, if/else
    - default assign!!!
  - genvar - like for loop, starts at 0
    - genvar i;
      generate
      for(i=0; i<=N; i=i+1) begin : blk (unique name)
      end
      endgenerate
  - use EECS register, don't use always @(posedge clk)
    - no register inference
    - use ce and rst!!!
- FPGA
  - MUX, latch, LUT (labeled)
  - has configurable logic blocks (CLB) "island" based
  - b/c between programs latches < vs LUT → FF → MUX
  - n LUT → 2^n entries!
  - Partitioning - draws circle how X inputs, 1 output
  - usually has dedicated blocks for b/c slow/expensive
  - combine LUTs → use MUX!

- Boolean Logic (circuit)
  - boolexpr - truth table - gate up
  - dual - swap OR/AND
    swap 0's/1's (CLEARED/SET)
  - SOP - sum of product, minterm
  - POS - product of sum, dual? maxterm
- K-map
  - gray code sides, 2 or more
  - circle largest grp, power of 2
    - can overlap!
    - each is product term
  - POS version - swap 0's and 1's
    - sum instead of product
  - cond delay, tradeoff
  - Demorgan's law!!!
- FSM
  - diagram table
  - Moore: output based on state
  - Mealy: output includes inputs
    - faster, difficult diagram
    - async output
    - fewer states
    - issue: if input glitches, bad

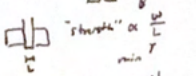
Moore Mealy



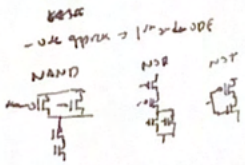
- register at end to be Moore timing
- State encoding
  - fewest FFs → # state = $2^{\#FF}$
  - easier dbg → one hot state encoding
    - 1 hot per state - good for FPGAs
- Verilog no can put default value
  - last one is used
- CMOS
  - MOSFET NMOS

  - inverter in $V_g = V$
  - lots of wire for manufacturing
    - accounts for imperfection
  - mask UV hardened
    - HF dissolve p^+ except mask
  - mask to implant n+
    - greater possible gate
    - still stripped!
  - grow oxide/strip
  - etch holes
  - metallize
  - etch again
  - FINFET
  - Gate all around

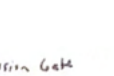
planar process



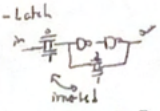
  - source is one more rail
  - model as ⏦
  - off current?
    I_{leak} V_T bit T_{ox} diff
    worse
  - use approx → linear model



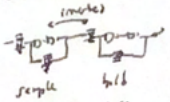
- Complex Gate
  - Pull up/down net (PUN, PDN)
  - duals of each other!
    - NOT between
  - Rail → PUN → dual for PDN
    ↑ opposite of the
    timing
    parameters
- Transmission Gate



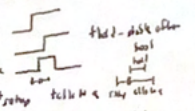
  - more real relations
- Tri-state buffer - 0, 1, Z
- Latch



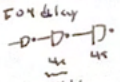
- 2 latch = flip flop



clock low = store
high = output

- Timing
  - CL time
  - 

  - usually setup + CL + clk→Q
  - FO4 delay



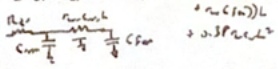
  - reliable as αN spikes if t_{RC} is small
  - Retiming
    extend loop first!
    - push thru gate
    - push around loops
    - combine!
    insert loop - try not to
    - combine
    - break the gate? need check
  - Delay Model
    - $C_g = TC_g$ $\gamma \approx 1$ usually

$$t_p = 0.69(3\tau) R_N C_G$$

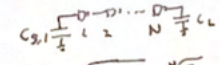
    no dependence!

$$f = \frac{C_L}{C_{in}}$$

    - inverter $t_{p0}(1 + f/\beta)$
      $C_{in} = 3WC_g$
      (scales from min)
    - wire $0.38 r_{wire} c_{wire} L^2$
    - $0.69 R_{dr}(C_w + C_{in}) + 0.69(R_{wire} + r_{wire} C_{fan}) L + 0.38 r_{wire} c_{wire} L^2$



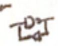
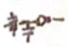
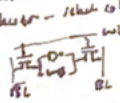
  - Large load



$$f = \sqrt[N]{C_L / C_{g,1}} = \sqrt[N]{F}$$

$$f = e^{\left(1 + \frac{\gamma}{f}\right)}$$

Matthew Tran
EE151
Final

- RISC-V
  - Pipeline
    - hazard - next instr dep on past
    - data hazard - reg not yet written
    - control hazard - branch, jump
    - forward
    - stall
- Power
  - performance, size, weight, heat
  - servers - Total Cost of Ownership (TCO)
  - Switching Energy CMOS
    - $\frac{1}{2}CV^2$ loss, independent of tech
    - reduce # clock transitions
    - reduce V_{dd} - limits clock
    - fewer transistors
    - reduce C per node - scale process
  - Dynamic Power
    $$P = \frac{1}{2}\alpha C V_{dd}^2 F$$
    (α: activity ~0.4)
  - Short circuit current loss when both on (lower clk)
  - leakage current - continuous, but lower I_{on}
    - raise I_{on} → ↑ V_{dd} or ↓ V_t or doping (↓ I_{off})
  - $P_{total} = P_{switch} + P_{sc} + P_{leak}$
  - Techniques for Low Power
    - Parallelism, Pipelining - split block up
      $P \propto F \cdot V_{dd}^2$
      - ↓ V_{dd}, ↓ F but same performance
        $F \propto V$ roughly
      - more cores also works
      - Scale V and F (DVFS)
        - better heat and/or energy
    - Power Down Idle
      - sleep transistors - better leak
        - slows clock rate
    - Slow down "slack paths"
      - most logic "too fast"
      - use multiple V_{dd} for noncritical path
      - or multi V_t (high V_t → slower)
    - Clock gating
      - use clock enable only if state change
    - Thermal Management
      - keep cool
- Memory
  - Volatile - needs power
    - Static - feedback 
    - dynamic - charge 
  - generic architecture
    - word line - select row
    - bit line - data to edge
    - core aspect ratio - keep ~1
    - address bits divided to row/col
    - row decoder - select word
    - col decoder - select col
  - SRAM
    - CMOS 
      ← requires correct sizing
    - read
      - precharge BL to V_{DD}
      - WL HIGH
      - cell pulls 1 down
      - measure w/ differential
    - write
      - diff drive bit line
      - size to overpower pullup
  - Decoder
    - predecoder - 2 bit = 4 signals
      - fewer stages, works nicely
    - col decoder is big MUX
      - pass transistor
  - DRAM
    - special process for big C ← stacked cap
    - 3T - non destructive read, avoid process need
    - 1T - need write after read, need refresh
    - latch based sense amp
      - init to EQ equal
      - then enable BL
      - then measure
- Multi-port Memory
  - add port
    - add another set of peripheral logic
- Combining Memory Blocks
  - tile smaller blocks "leafs"
  - leaf size limited by RC delay
  - low power by activate banks
  - delay & energy mostly from IO interconnect
  - larger mem? (bigger words)
    just split the data
  - larger mem? (bigger addr)
    use MUX w/ last bit
  - add read port?
    write to both
  - add write port?
    MUX w/ "residency" (which block recent)
    - write same addr - pick policy
- FIFO - queue
  - 2 ptr
  - full, empty when ptr match
- Cache
  - temporal, spatial locality
  - Tag, Index, Byte select
  - parallel tag = { fully associative - no index; set associative - N entry each index }
  - direct
  - replacement policy
- Parallelism
  - associativity of log n lower bound
    - trees
    - higher hardware cost
  - Time Multiplex
    - ↓ cost ↑ time
  - Loop
    - better throughput, worse latency
    - pipelining - limited by FF speed
  - Pipeline limit
    - clock skew
    - unequal stages
    - FFs dominate cost
    - clock distribution
    - hazards - feedback "loop carry dependency"
      - reorder operation - shorten feedback
      - C-slow
        - run n-indept streams in parallel
        - replace each reg w/ n regs
        - lower F per channel, but n of them
          - same throughput
        - MUX streams
        - multithread uses this
      - SIMD
- RT Language
  - ; sep cycles / , same cycle
  - FSM follow directly
  - Design Process
    - Problem Spec
    - "Algorithm"
    - Micro-architecture
    - Cost, Perf, Power Analysis
    - Optimization, Variation
    - Detailed Design
  - Resource Utilization Chart
    - show per resource
  - Modulo Scheduling
    - pick characteristic section
    - schedule over time wrap, +1
    - if not fit, increase size
- Adder
  - Carry Ripple - $T \propto n$, cost $\propto n$
  - subtraction just XOR
    - 2's complement, C_{in} is 1
  - Carry Select - do both, select 1
    - multiple ripple stages
    - $\sqrt{N}$ stages, $\sqrt{N}$ bits if equal
    - $T \propto \sqrt{N}$, cost $= 2 \times$ ripple + mux (2x cost)
    - N = 3 better
    - if unequal, can fiddle to better match delays
  - Carry Look Ahead (CLA) O(log n)! 4x cost
    - propagate - $p_i = a_i \oplus b_i$
    - generate - $g_i = a_i b_i$
    - group result - $C_{out} = G + PC_{in}$
      $S_i = P_i \oplus C_i$
      $P = P_1 P_2$
      $G = G_2 + G_1 P_2$
      $C_{i+1} = g_i + p_i C_i$
    - associative!
    - parallel prefix adder - how to group
      - Kogge-Stone
      - Ladner-Fischer
      - Brent Kung
      - Han Carlson
    - mix w/ ripple, use CLA to make carry
  - bit serial adder
    - O(1) hardware, O(n) cycles
- Multiplier
  - Shift and Add
    - add partial products
      - sign extend
    - cost $\propto n$, $T \propto n$ cycles
  - Array multiplier
    - large, n^2 cost
    - Carry save
      - 3 bit in, 2 out
      - need CLA at end, full adder
      - trees! Wallace Tree multiplier
  - Booth recoding
    - group of 2 bits, half delay!
  - Bit serial
    - $O(n^2)$ time, cost $\propto$ 1
  - Baugh Wooley - signed mult
    - add constant to sign bit, no sign extend
    - add constant at end
    - same hardware cost as unsigned
  - Constant Coefficient (KCM)
    - shift and add idea
    - use subtraction
    - Canonical Signed Digit (CSD)
      $15 \to 10\bar{1}$ (mmm $2^n - 1$ basically)
      $1\bar{1} \to 01$ (mmm $2^n - 2^{n-1} = 2^{n-1}$)
    - do factorization to simplify further
      - exhaustive search
- Shifter / Rotator
  - O(log n) of power of 2 shifts
  - barrel shifter - n^2 cost
    - crossbar switch
    - any permutation, rotate, etc
    - $N \log N$ control
- Clock Nonidealities
  - skew - time diff between edges +/-
    - jitter - random variation
      short term t_{jc}
      long term t_{jc}
  - skew - many ← crit path
  - neg skew - affect min cycle time
  - pos skew - hold time
- Clock Distribution
  - H-tree equal length & buffer
- Chip Packaging
  - bond wires on pads - limited
  - flip-chip - distributed pad
    - solder ball on pad (more pads)
  - ESD - diodes + guard rings

- Power Distribution
  - IR drop, L, metal resistance
  - most give power, thick metal layer
  - metal migration
    - current ↑ min metal
  - $L\frac{dI}{dt}$ drop, IR drop
  - usually 5-10% PSU variation
    - flip chip better
  - decoupling caps
- Faults
  - Design
  - Manufacturing
  - Runtime
  - Modeling
    - ATPG - test vector
    - BIST - self test
    - scan chain
    - extra hardware sacrificial
      - laser fix
  - Runtime
    - permanent, transient, intermittent
  - Physical Fault
    - NBTI, PBTI } ↑ V_t, ↓ mobility
    - HCI }
    - TDDB - shorts
    - electromigration
- ECC
  - parity
  - Hamming code
    - SEC - single error correct
    - DED - double error detection (P is whole thing)
    - bits at powers of 2 are extra
      - protect bits even parity
    - error tells where it is!
      C=0, P=0 no error
      C!=0, P=1 SEC
      C!=0, P=0 DED
      C=0 P=1 error in P