Energy and Power

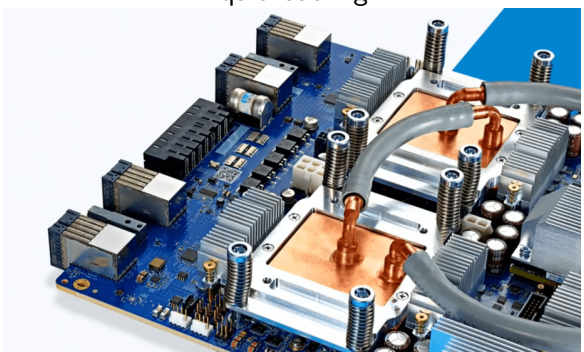
Nachiket Kapre
nachiket@uwaterloo.ca

Outline

- Why care about power/energy?
- Physical Equations of Power/Energy
- Tradeoffs
  - Voltage-Frequency Scaling
  - Parallelism, Pipelining, and Time-Multiplexing
  - Activity Rates and associated analysis
  - Clock gating and Power gating

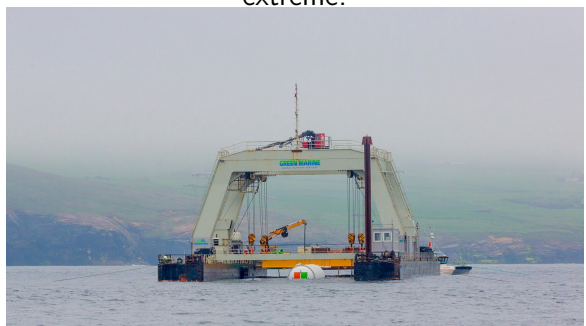
Google Liquid-Cooled TPU

TPU v3 runs very hot, consumes too much power! Cannot be cooled with fans, needs liquid cooling!



MSR Underwater Datacenter

Microsoft Research put a small datacenter under the Scottish sea. Liquid cooled in the extreme!



Why do we care about Power/Energy of Computing Chips?

- Datacenter power usage now 10% of global demand and rising
- Desktops/servers constrained by how quickly they can be cooled $\rightarrow$ Practical capability 100 W/cm^2
- Mobile phones, IoT/embedded devices have limited battery capacity

- **Datacenter**: TCO (Total Cost of Ownership) entails buying more expensive silicon if electricity + cooling costs will stay low
- **Power Limits**: *Dark Silicon*, where we have to keep parts of our large multi-billion transistor chip idle or it will get too hot/melt
- **Energy Limits**: Cellphones are limited by battery capacity which is a measure of energy. Must design chips differently.

Reducing Power and Energy?

- Reducing power $\rightarrow$ reduce activity (do fewer things per unit time)
  - Implication on reduced cooling/heat removal costs
- Reduce energy $\rightarrow$ efficiency (do fewer things, period!)
  - Implication on extending battery life

Textbook Definitions

- Power is rate of consumption of energy (Energy/Time)
- $P \propto f \times C \times V^2$
  - P = Power
  - f = Frequency
  - C = Capacitance
  - V = Voltage
- Energy is the amount of work that can be accomplished by the circuit.
- $E \propto C \times V^2$

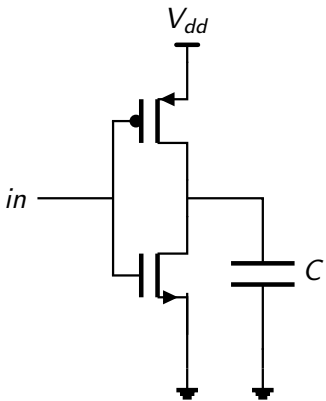
Reducing Power

- $P \propto f \times C \times V^2$
- To reduce P, we can reduce:
  - f = frequency of the chip, slow it down?
    - Reduce f will also delay time required to complete task!
  - C = capacitance of the design
    - Reduce c requires reducing circuit size, using right kind of hardware modules, novel materials
  - V = voltage
    - Reduce V provides a *quadratic* reduction in power. Most effective technique! But cannot drop V too low, and it affects f (see next slide)
- $E \propto \times C \times V^2$
- To reduce E, we can reduce C or V (see above)

Power Equation Breakdown

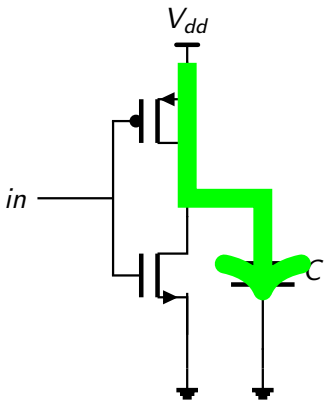
- Possible to decompose P into its constituent contributors
  - $P = \alpha \times f \times C \times V^2$
- $P = P_{dynamic} + P_{static}$
  - $P_{dynamic}$ = Power due to switching activity in the circuit
  - P_{static} = Power due to leakage currents in the circuit (chip will draw power even if not doing anything useful)
- $P_{dynamic} = P_{switching} + P_{short-circuit}$
  - $P_{switching}$ = Dynamic power due to useful work in the design
  - $P_{short-circuit}$ = Dynamic power due to mismatched rise/fall times leading the momentary short circuits between V_{dd} and GND.

$P_{switching}$



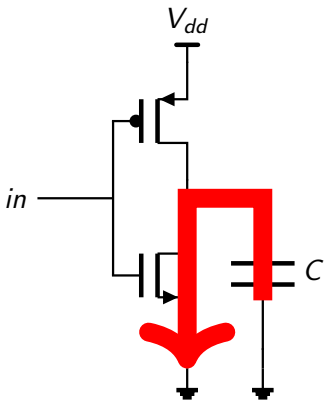
- The $0 \rightarrow 1$ **charging** path goes through the PMOS
  - Energy required to charge capacitor is $\frac{1}{2} \times C \times V_{dd}^2$
- The $1 \rightarrow 0$ **discharging** path goes through the NMOS
  - Energy required to discharge capacitor is $\frac{1}{2} \times C \times V_{dd}^2$
- Activity Factor α = Number of Transitions / (Number of Signals $\times$ Number of Cycles)
- $P_{dynamic} = \alpha f \times C \times V_{dd}^2$

$P_{switching}$



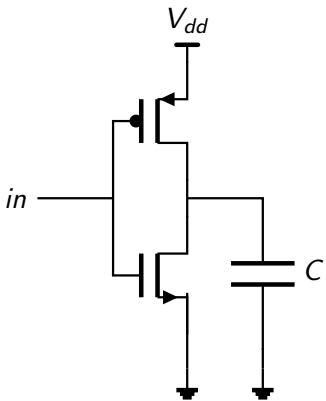
- The $0 \rightarrow 1$ **charging** path goes through the PMOS
  - Energy required to charge capacitor is $\frac{1}{2} \times C \times V_{dd}^2$
- The $1 \rightarrow 0$ **discharging** path goes through the NMOS
  - Energy required to discharge capacitor is $\frac{1}{2} \times C \times V_{dd}^2$
- Activity Factor α = Number of Transitions / (Number of Signals $\times$ Number of Cycles)
- $P_{dynamic} = \alpha f \times C \times V_{dd}^2$

$P_{switching}$



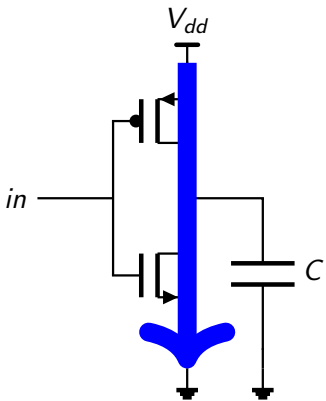
- The $0 \rightarrow 1$ **charging** path goes through the PMOS
  - Energy required to charge capacitor is $\frac{1}{2} \times C \times V_{dd}^2$
- The $1 \rightarrow 0$ **discharging** path goes through the NMOS
  - Energy required to discharge capacitor is $\frac{1}{2} \times C \times V_{dd}^2$
- Activity Factor α = Number of Transitions / (Number of Signals $\times$ Number of Cycles)
- $P_{dynamic} = \alpha f \times C \times V_{dd}^2$

$P_{short-circuit}$



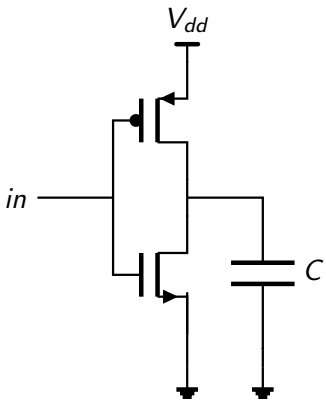
- Short circuit between V_{dd} and *GND* happens briefly during slow transitions at *in*
  - For brief periods, both PMOS and NMOS transistors will be turned ON
  - A direct path is established between the power supply and ground
  - This is wasted current that is a tax on operation
- $P_{short} = \alpha f \times t_{short} \times I_{short} \times V_{dd}$

$P_{short-circuit}$



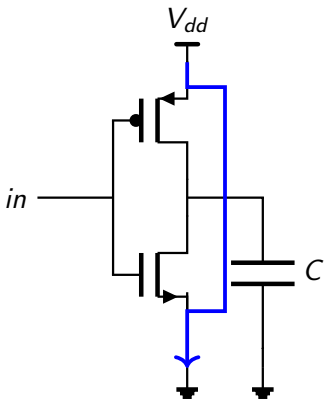
- Short circuit between V_{dd} and *GND* happens briefly during slow transitions at *in*
  - For brief periods, both PMOS and NMOS transistors will be turned ON
  - A direct path is established between the power supply and ground
  - This is wasted current that is a tax on operation
- $P_{short} = \alpha f \times t_{short} \times I_{short} \times V_{dd}$

P_{static}



- Parasitic diode in the MOS substrate layer
- Constant leakage current flows between V_{dd} and GND throughput circuit operation
- This current is independent of circuit activity and is hence termed, static/leakage current!
- Leakage Current $I_{leak} = e^{\frac{-q \times V_t}{k \times T}}$
- $P_{static} = I_{leak} \times V_{dd}$

P_{static}



- Parasitic diode in the MOS substrate layer
- Constant leakage current flows between V_{dd} and GND throughput circuit operation
- This current is independent of circuit activity and is hence termed, static/leakage current!
- Leakage Current $I_{leak} = e^{\frac{-q \times V_t}{k \times T}}$
- $P_{static} = I_{leak} \times V_{dd}$

Voltage Scaling

- Voltage reduction is an interesting approach here because:
  - Lower both power and energy by a quadratic factor
  - Lower voltage forces slower operation $f \downarrow$
- Alpha Power Law of MOS: $\frac{1}{f} = t = K \times \frac{V_{dd}}{(V_{dd} - V_{th})^a}$
  - Approximate $a = 2$ and $V_{th} = 0$, which yields $t = K \times \frac{1}{V_{dd}}$
- If we lower V_{dd}, t increases inversely $\rightarrow$ circuit slows down

Energy-Delay Product

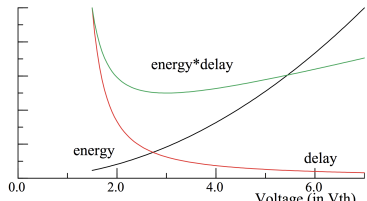


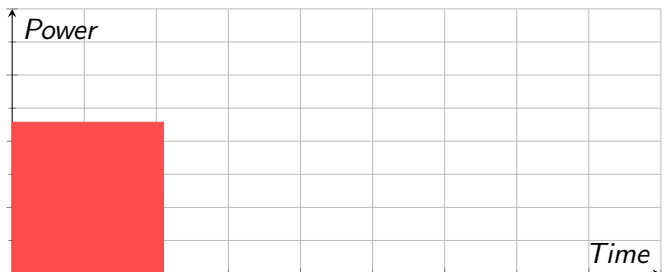
Figure 1. Energy and Delay vs. Voltage

Low-Power Digital Design, Horowitz et al

- Energy ($E \propto C \times V_{dd}^2$) can be lowered by lowering V_{dd}
- However, Delay $t = K \times \frac{1}{V_{dd}}$ also increases with lower V_{dd}
- Thus, you can always lower energy if you slow down the computation
  - Hence, need new metric of efficiency $\rightarrow$ Energy-Delay product

Frequency Scaling

- Scaling frequency f can reduce power, but will not affect energy
- **Faster** Frequency will get the job done sooner $\rightarrow$ Can turn off chip?
- **Slower** frequency will never draw more than specified power $\rightarrow$ battery current draw will be capped



Frequency Scaling

- Scaling frequency f can reduce power, but will not affect energy
- **Faster** Frequency will get the job done sooner $\rightarrow$ Can turn off chip?
- **Slower** frequency will never draw more than specified power $\rightarrow$ battery current draw will be capped

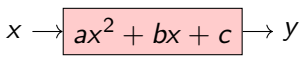
Contemporary scandal

- Apple throttling old iPhones → `https://ifixit.org/blog/11208/batterygate-timeline/`
- Cause: Unexpected shutdowns due to battery degradation
- Fix: Slow down iPhones to lower burden on battery
- Result: Worldwide scandal → low-cost battery replacement apology.

Parallelism, Pipelining, Time-Multiplexing (Scheduling)

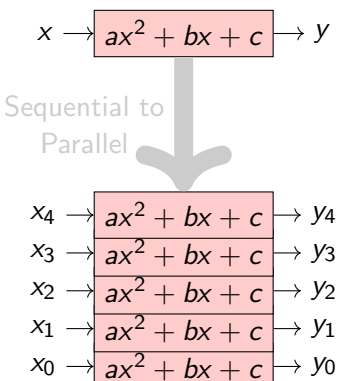
- Concurrency helps us reduce energy! Counter-intuitive idea, but powerful
- Concurrency can be delivered by either:
  - Parallel copies of hardware
  - Pipelining of hardware
- Conversely, time-multiplexing increases energy due to sequentialization of evaluation!
- If you throw in voltage scaling, we need a different metric (Energy-Delay^2) $\rightarrow$ ED^{2}P metric out of scope of 327

EDP of Parallel Hardware



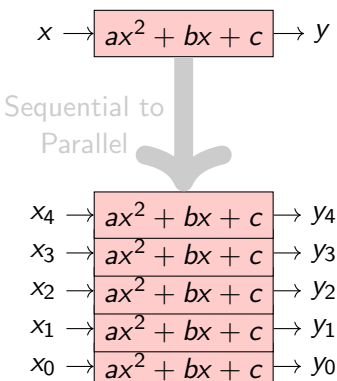
- Sequential Evaluation of $ax^2 + bx + c$. Output register y.
- You know that:
  - Energy/Evaluation of the polynomial datapath is E
  - Delay/Evaluation is D
- If circuit must compute on N inputs:
  - Total Energy $= N \times E$
  - Total Delay $= N \times D$
  - Energy-Delay Product $= N^2 \times E \times D$

EDP of Parallel Hardware



- Parallel Evaluation of $ax^2 + bx + c$
  - Assume distributing x, gathering y is free.
- P parallel copies of the hardware
  - Energy/eval/copy $= E$, and Delay/eval/copy $= D$
  - Thus, Energy/Eval $= P \times E$, and Delay/Eval $= D$ (because of parallelism!)
- N inputs will be distributed across P copies $\rightarrow$ each copy gets $\frac{N}{P}$
  - Total Energy $= \frac{N}{P} \times P \times E = N \times E \rightarrow$ Total energy stays unchanged!
  - Total Delay $= \frac{N}{P} \times D$
  - Energy-Delay Product $= N^2 \times E \times D \times \frac{1}{P} \rightarrow P\times$ better than sequential

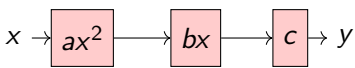
EDP of Parallel Hardware



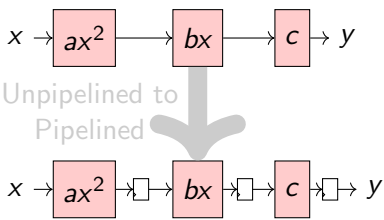
- If customer is satisfied with original throughput of single evaluation/cycle $\rightarrow$ P parallel evaluations is too fast?
- We can now reduce voltage to $\frac{V_{dd}}{P}$
  - Recall $t \propto \frac{1}{V_{dd}} \rightarrow$ Delay/eval/copy $D \times P$
  - Recall $E \propto V_{dd}^2 \rightarrow$ Energy/eval/copy $\frac{E}{P^2}$
- Total Delay $= \frac{N}{P} \times D \times P = N \times D$
- Total Energy $= \frac{N}{P} \times P \times \frac{E}{P^2} = N \times \frac{E}{P^2}$
- Energy-Delay Product $= N \times D \times N \times \frac{E}{P^2} = N^2 \times E \times D \times \frac{1}{P^2} \rightarrow$ P times better than parallel

EDP of Pipelined Hardware

- Pipelining is another way to improve energy efficiency of hardware
- **Intuition**: Hardware stays idle and consumes energy when not in use

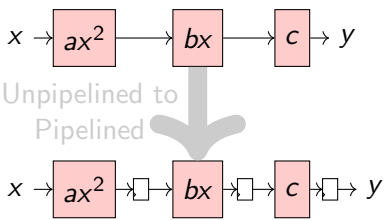


EDP of Pipelined Hardware



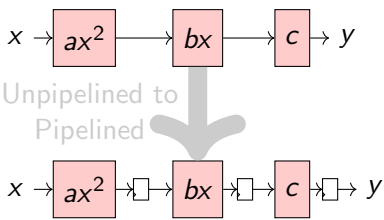
- Pipelining is another way to improve energy efficiency of hardware
- **Intuition**: Hardware stays idle and consumes energy when not in use
- Unpipelined processing:
  - Energy E_1,E_2, and E_3 for each step
  - Delay D_1, D_2, and D_3 correspondingly
  - Total Energy $= E_1 + E_2 + E_3$
  - Total Delay $= D_1 + D_2 + D_3$
  - EDP $= E_1 \times (D_1 + D_2 + D_3) +$ $E_2 \times (D_1 + D_2 + D_3) + E_3 \times (D_1 + D_2 + D_3)$
    - If all qty equal, EDP $= 3 \times E \times 3 \times D$

EDP of Pipelined Hardware



- Pipelining is another way to improve energy efficiency of hardware
- **Intuition**: Hardware stays idle and consumes energy when not in use
- Pipelined processing
  - Energy E_1,E_2, and E_3 for each step
  - Delay max(D_1,D_2,D_3) due to pipelining
  - Total Energy = $E_1 + E_2 + E_3$
  - Total Delay = max(D_1,D_2,D_3)
  - EDP= $(E_1 + E_2 + E_3) \times max(D_1, D_2, D_3)$
    - If all qty equal, EDP = $3 \times E \times D \rightarrow 3\times$ better than unpipelined

EDP of Pipelined Hardware



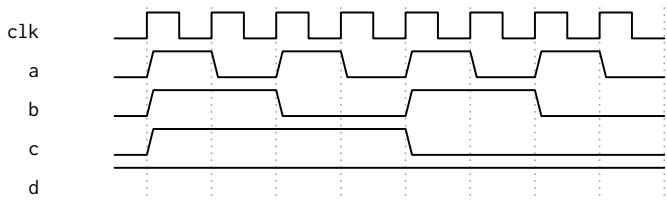
- While pipelining is great, it may be too fast for what you want
  - Unpipelined design delay $D_1 + D_2 + D_3 \approx 3 \times D$
  - Pipelined design delay $max(D_1, D_2, D_3) \approx D$
- To run pipelined design at $3 \times D$ delay, must scale V_{dd} by $\frac{1}{3}\times$.
  - Recall $t \propto \frac{1}{V_{dd}}$
  - Recall $E \propto {V_{dd}}^2$
- Thus, energy per stage will decrease to $\frac{E}{3^2}$, ...
- EDP $= (3 \times \frac{E}{3^2}) \times (3 \times D) = E \times D \rightarrow$ 3 times better than ideal pipelined scenario where V_{dd} didn't change

Activity Rate Analysis

- Activity rate α contributes to dynamic power $P_{dynamic}$.
- Recall $P_{dynamic} = \alpha f \times C \times V_{dd}^2$
- Thus, minimizing α is crucial $\rightarrow$ avoid unnecessary transitions
- Bit flips 0$\rightarrow$1 and 1$\rightarrow$0 will cost power.
- Simple RTL coding pattern to reduce bitflips:

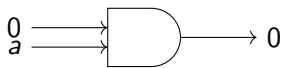
```
always@(posedge clk) begin
    if(valid=1'b1) then
      data <= input_data;
    end if;
    output_valid <= valid;
    output_data <= compute(data);
end;
```

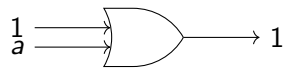
What is Activity Factor?



- Activity factor (α) is the rate of $0 \rightarrow 1$ or $1 \rightarrow 0$ signal transitions in the circuit
- α = (Num. of $0 \rightarrow 1$ + Num. of $1 \rightarrow 0$) / Tot. Cyc.
- Activity factor of a $\rightarrow \alpha_a = \frac{8}{8} = 1.0$
- Activity factor of b $\rightarrow \alpha_b = \frac{4}{8} = 0.5$
- Activity factor of c $\rightarrow \alpha_c = \frac{2}{8} = 0.25$
- Activity factor of d $\rightarrow \alpha_d = \frac{0}{8} = 0$

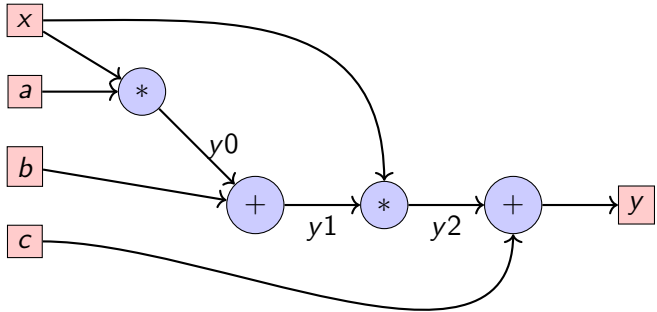
Activity Analysis – Simple gates





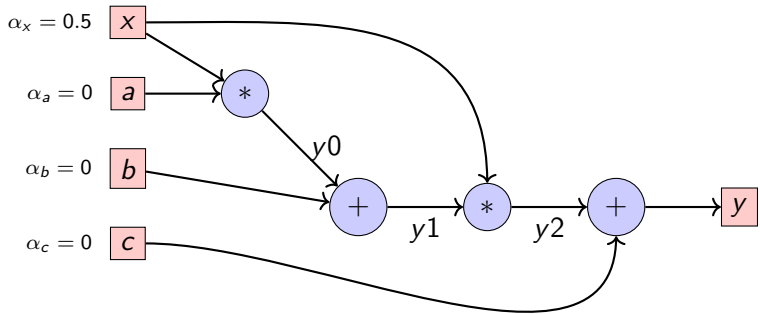
- When one of the inputs to an AND or OR gate is an appropriate constant, the output gets locked to a constant!
- AND gate, either input is 0, output is 0
- OR gate, either input is 1, output is 1
- If you know this, you can compute activity rate for the output as 0 upfront!

Activity Analysis – Complex operations



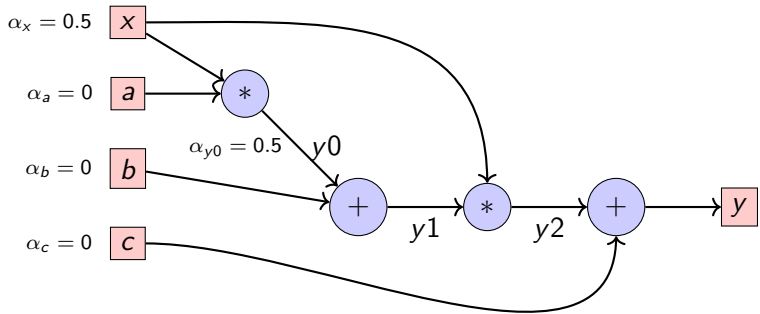
- If you only know input activity rates, you have to assume worst case and sum up the rates of both inputs.
- If one of the inputs is a constant, then output rate = input rate of the changing input
- However, this exaggerates activity rates and a simulation-based methodology is typically used.

Activity Analysis – Complex operations



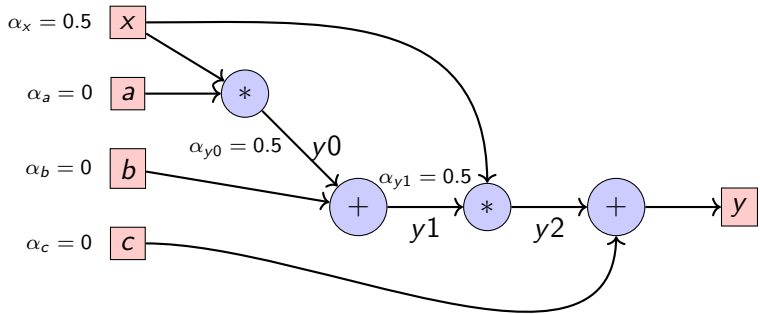
- If you only know input activity rates, you have to assume worst case and sum up the rates of both inputs.
- If one of the inputs is a constant, then output rate = input rate of the changing input
- However, this exaggerates activity rates and a simulation-based methodology is typically used.

Activity Analysis – Complex operations



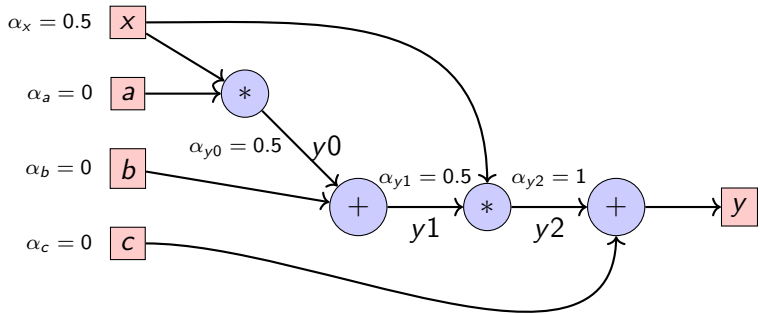
- If you only know input activity rates, you have to assume worst case and sum up the rates of both inputs.
- If one of the inputs is a constant, then output rate = input rate of the changing input
- However, this exaggerates activity rates and a simulation-based methodology is typically used.

Activity Analysis – Complex operations



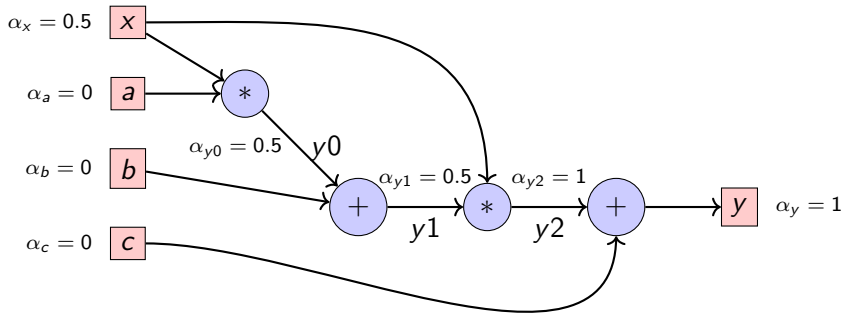
- If you only know input activity rates, you have to assume worst case and sum up the rates of both inputs.
- If one of the inputs is a constant, then output rate = input rate of the changing input
- However, this exaggerates activity rates and a simulation-based methodology is typically used.

Activity Analysis – Complex operations



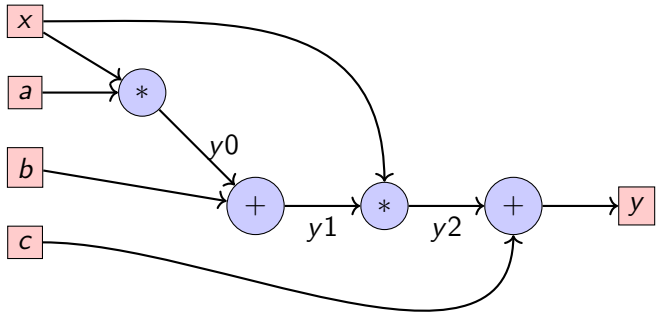
- If you only know input activity rates, you have to assume worst case and sum up the rates of both inputs.
- If one of the inputs is a constant, then output rate = input rate of the changing input
- However, this exaggerates activity rates and a simulation-based methodology is typically used.

Activity Analysis – Complex operations



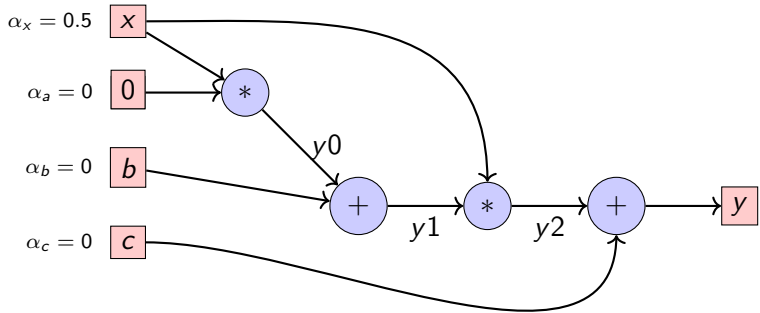
- If you only know input activity rates, you have to assume worst case and sum up the rates of both inputs.
- If one of the inputs is a constant, then output rate = input rate of the changing input
- However, this exaggerates activity rates and a simulation-based methodology is typically used.

Activity Analysis – Complex operations (Input values)



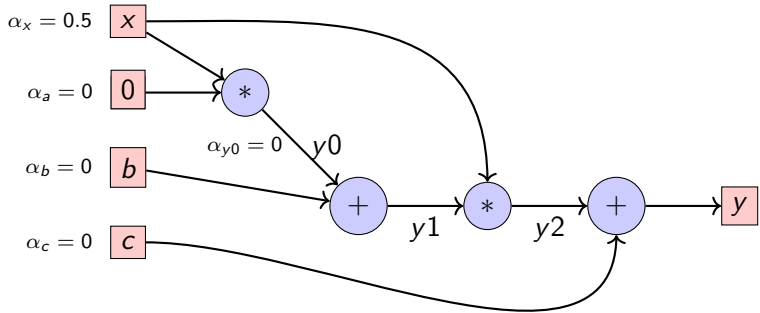
- Knowing the value of input can help significantly improve bounds on α
- Simulations are the best strategy for tracking what's happening on each signal y_1, y_2 and y to compute exact activity rates
- Simulations must be rigorous enough to model overall behavior

Activity Analysis – Complex operations (Input values)



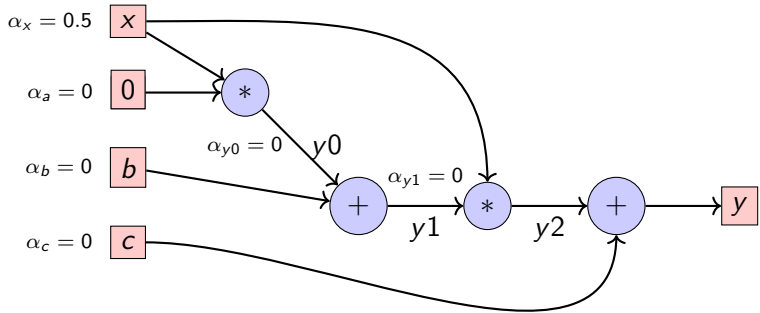
- Knowing the value of input can help significantly improve bounds on α
- Simulations are the best strategy for tracking what's happening on each signal y_1, y_2 and y to compute exact activity rates
- Simulations must be rigorous enough to model overall behavior

Activity Analysis – Complex operations (Input values)



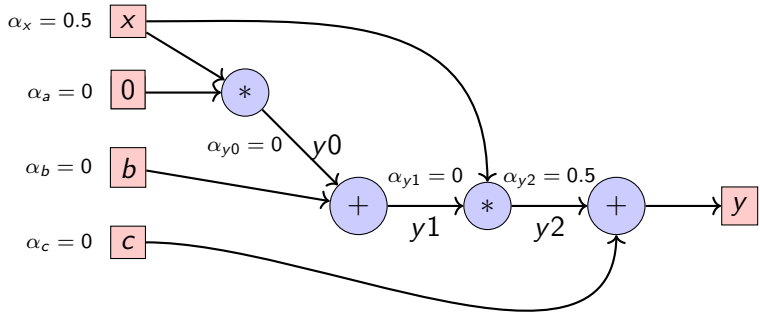
- Knowing the value of input can help significantly improve bounds on α
- Simulations are the best strategy for tracking what's happening on each signal y_1, y_2 and y to compute exact activity rates
- Simulations must be rigorous enough to model overall behavior

Activity Analysis – Complex operations (Input values)



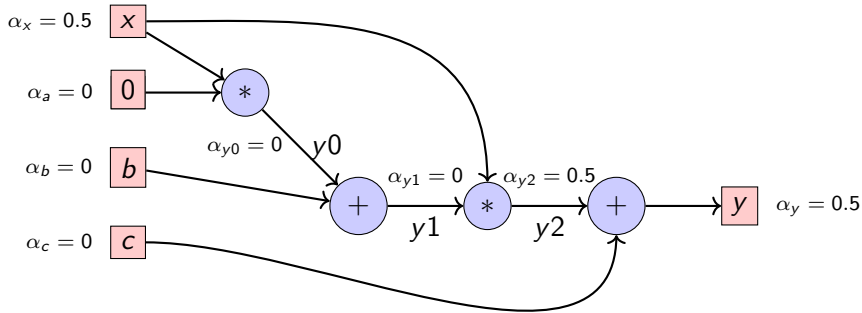
- Knowing the value of input can help significantly improve bounds on α
- Simulations are the best strategy for tracking what's happening on each signal y_1, y_2 and y to compute exact activity rates
- Simulations must be rigorous enough to model overall behavior

Activity Analysis – Complex operations (Input values)



- Knowing the value of input can help significantly improve bounds on α
- Simulations are the best strategy for tracking what's happening on each signal y_1, y_2 and y to compute exact activity rates
- Simulations must be rigorous enough to model overall behavior

Activity Analysis – Complex operations (Input values)



- Knowing the value of input can help significantly improve bounds on α
- Simulations are the best strategy for tracking what's happening on each signal y_1, y_2 and y to compute exact activity rates
- Simulations must be rigorous enough to model overall behavior

Class Wrapup

- Energy and Power use are fundamental to circuit design
- Dataceneters take up $\approx$10% of total electricity demand worldwide and growing $\rightarrow$ global warming forces us to take this seriously
- Power increasingly dominated by static components $\rightarrow$ power gating tricks necessary
- Dynamic components depend on activity rates $\rightarrow$ clock gating is important here
- Pipelining and parallelism are key to unlocking energy efficiency