IEP on Mapping DSP Algorithms to Architectures

Note Title 2/22/2010

Lecture 1: Nitin Chandrachoodan
- Lab: FIR filtering

Lecture 2 (Feb 23): Dr. T. G. Venkatesh

Lecture 3 (Feb 24): Dr. C. P. Ravikumar

Lecture 4 (Feb 25): Dr. K. Sridharan
Dr. A. Ravishankar

5 (Feb 26). Dr. Shankar Balachandran

Outline

- DSP applications, properties
- Architectures - HW, SW, mixed
- Fundamental limit on throughput
- Number representations
- Computer-Aided Design (CAD) / Complexity Theory
- Parallelism and Pipelining

DSP applications

- Filtering (FIR, IIR) : $y(n) = \sum a_k y(n-k) + \sum b_k x(n-k)$
- Transforms (FFT, DCT) : $X(k) = \sum x(n) e^{-j2\pi kn/N}$
- Decomposition (SVD, LU, QR): Matrix operations

- Dot product / Matrix-vector operations
- Arithmetic operations: multiplication, addition / shift operation
- Sample time - depends only on application
- Non-terminating
  - dataflow

Implementation

- Hardware / Software
  - Multiply, add, shift register in hardware
  - Custom VLSI circuits : ASIC
    Programmable Logic : FPGA
  - Software : Programmable processors
    - DSP specific extensions - MAC, bit reversed addressing
    - Fixed point vs. Floating point

Number representation.

- Binary number system.
  - Integers eg. 0110 ⟶ 6 (decimal).
    Two's complement 1010 ⟶ −6 (decimal)
    (assuming 4-bit 2's complement).
  - Fractional values
    - Divide by 2^{n-1} ⟶ range becomes $[-1, +1)$.

n-bits: $[-2^{n-1}, 2^{n-1}-1]$

8-bits: $[-128, +127]$.

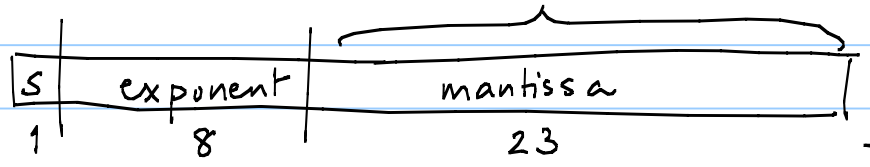
Fractional: $(-1, +127/128)$

Resolution: $1/256$ of range: $[-1, +1)$.

32-bit numbers: $[-1, +1) \rightarrow \frac{1}{2^{31}}$ smallest difference

Floating point numbers.

32-bits



→ IEEE FP (single precision).

Number magnitude = $(1.\text{mantissa}) \times 2^{\text{exponent}-127}$

Smallest +ve : $(1.000\ldots0) \times 2^{-126}$

Largest +ve : $(1.111\ldots1) \times 2^{+127} \simeq 2^{128}$

Much higher <u>dynamic range</u> than fixed point.

$a = 1.23 \times 10^{-5} \longrightarrow 1.23 \times 10^{-5}$

$b = 4.5 \times 10^{2} \longrightarrow 450\,00000 \times 10^{-5}$

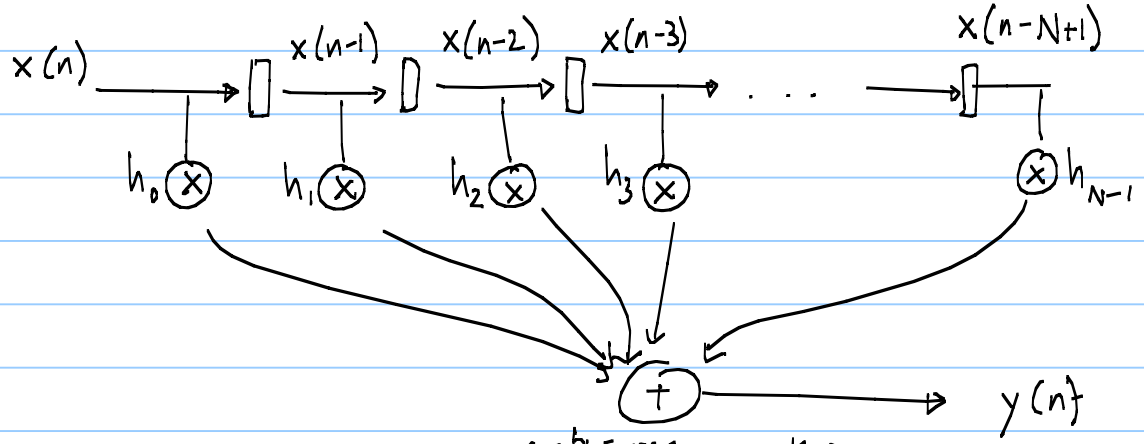
$450\,00001.23 \times 10^{-5} \longrightarrow 4.500000123 \times 10^{2}$

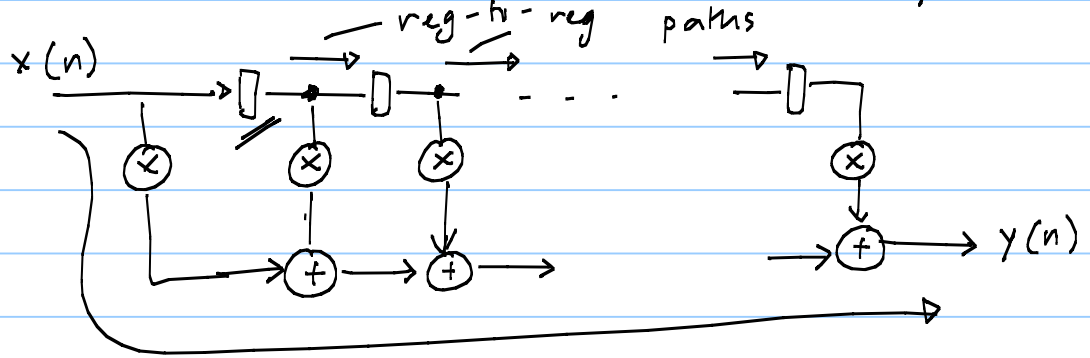
Implementation Flow

- High level description - Specifications
  - Equations
  ] Matlab, C/C++
- Architecture
  - Building blocks, speed and area
  - Block diagram
  - Optimizations : # bits, # coefficients
  ] Hardware Description Lang
  Verilog, VHDL
- Compile (synthesize).
  - Mapping to gates
  - Place & Route

Implementation of FIR filters.

$$y(n) = \sum_{k=0}^{N-1} h_k x(n-k).$$





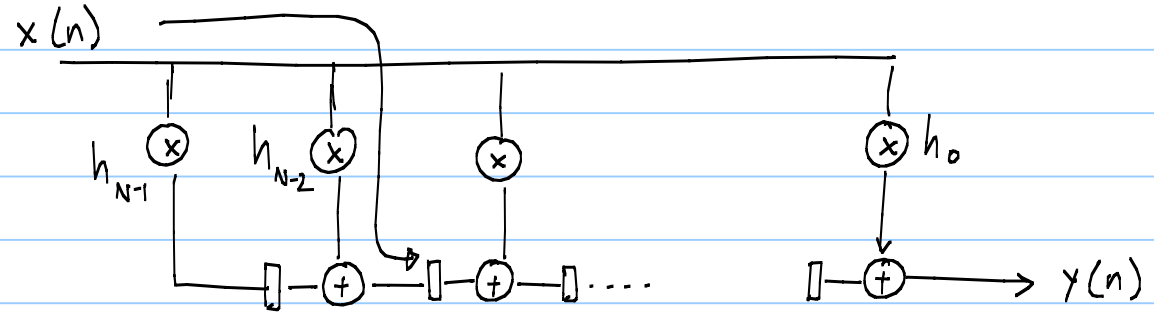
Direct form - I.

Area of filter: N-1 registers
N multipliers
N-1 adders

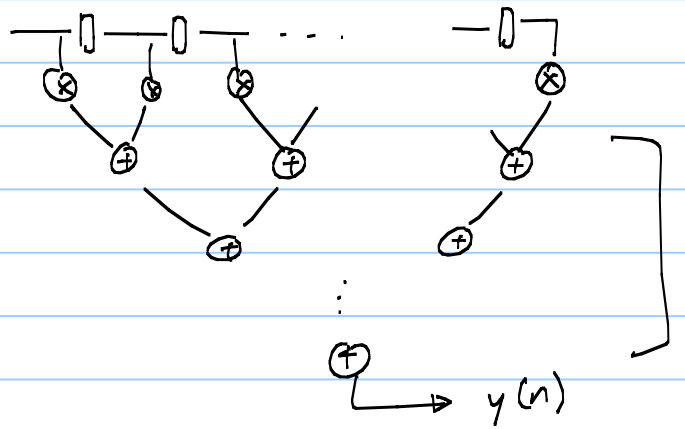
Time: Critical path: 1 multiplier, N-1 adders.

$= T_m + (N-1) T_a$.

Broadcast structure



Critical path: $T_m + T_a$.



N values to be added.

→ N/2

→ $N/2^2$

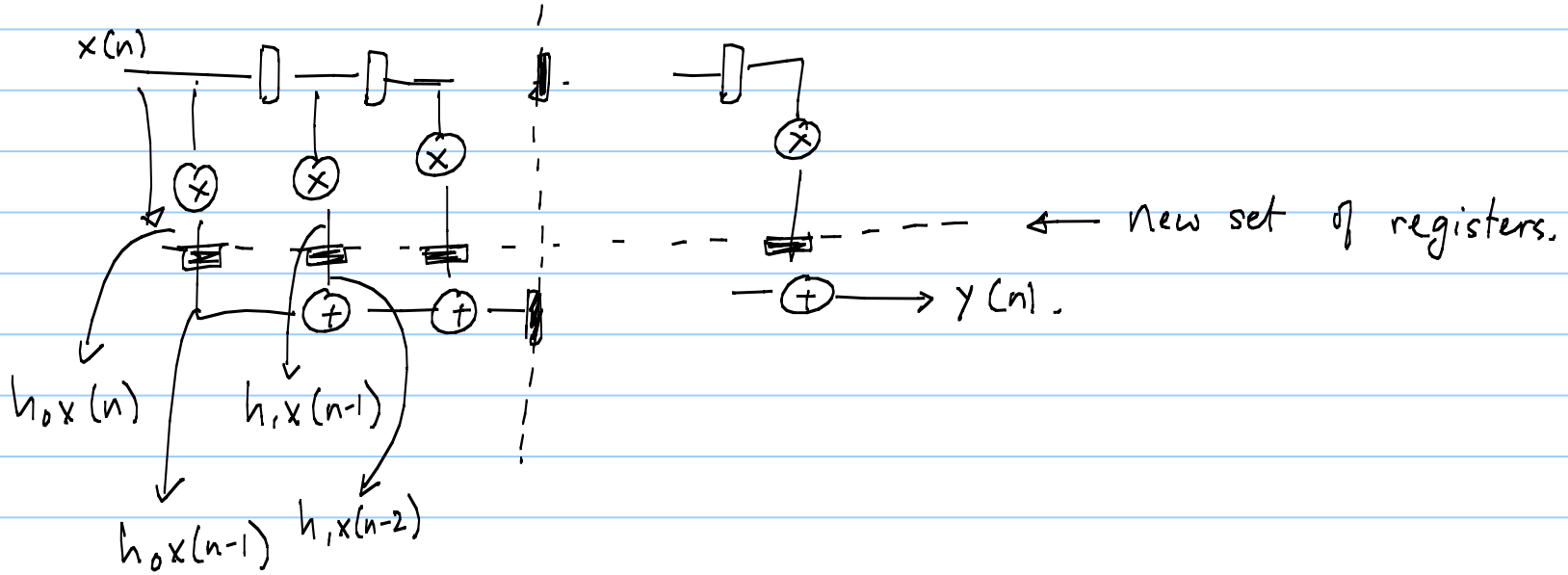
→ $N/2^3$

⋮

stages $2^K = N$

$K = \lceil \log_2 N \rceil$

$CP = T_m + \lceil \log_2 N \rceil T_a$.



New $y(n) = h_0 x(n-1) + h_1 x(n-2) + \dots + h_{N-1} x(n-N)$

Old $y(n) = h_0 x(n) + h_1 x(n-1) + \dots + h_{N-1} x(n-N+1)$

CP: either T_m or $(N-1)T_a$

More registers → can bring CP down to $\max(T_m, T_a)$.

Is this max (T_m, T_a) a fundamental limit.

- FIR filter, coeffts are known.
- Type of hardware known: T_m, T_a, registers.

Fundamental limit on Throughput

eg. FIR filter.

$$y(n) = \sum h_k x(n-k).$$

For each $y()$, N multiplications

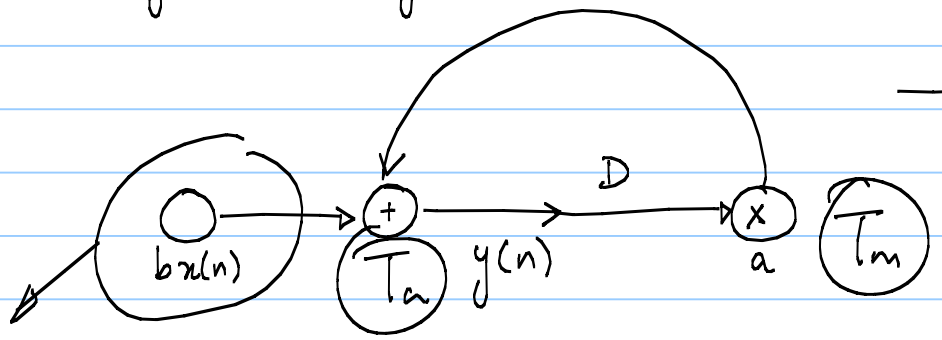
Assuming all input samples are readily available

$$y(0) = \sum h_k x(0-k) \longrightarrow N \text{ mult}, N-1 \text{ additions}$$
$$y(1) = \sum h_k x(1-k) \longrightarrow \text{"} \quad \text{"}$$
$$y(2) = \sum h_k x(2-k) \longrightarrow \text{"} \quad \text{"}.$$
$$\vdots$$

$\Rightarrow$ with enough parallelism, any throughput can be achieved.

Recurrence equations

$$y(n) = a\,y(n-1) + b\,x(n)$$



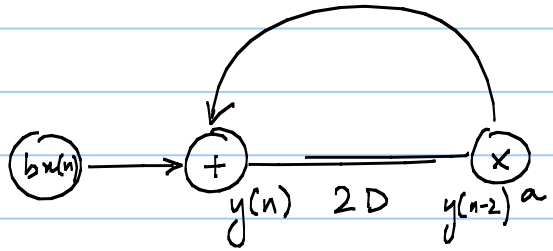
→ Cyclic graph.

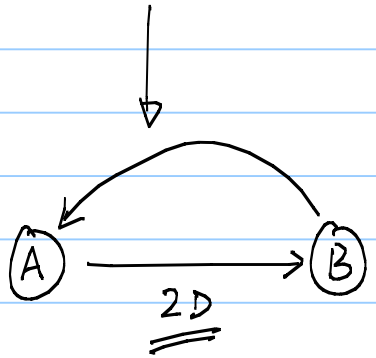
$y(n)$ only after $y(n-1)$ → mult by a → added to $bx(n)$

Given $y(n-1)$, minimum time to get $y(n)$ = $T_m + T_a$

$y(n) = a\,y(n-2) + b\,x(n)$

$y(2) \leftarrow y(0)$
$y(3) \leftarrow y(1)$





Invocation / Firing

A_0	B_0
A_1	B_1
A_2	B_2
A_3	B_3
⋮	⋮

B_0 depends on A_{-2}
B_1 depends on A_{-1}

$B_0 \rightarrow A_0 \rightarrow B_2 \rightarrow A_2 \rightarrow B_4 \rightarrow A_4 \ldots$
$B_1 \rightarrow A_1 \rightarrow B_3 \rightarrow A_3 \rightarrow B_5 \ldots$

Time for one complete cycle : $T_B + T_A$

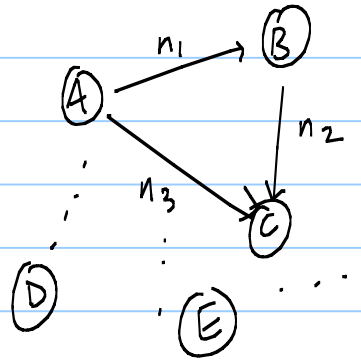
In one such cycle, how many samples : 1 on each stream ($B_0 - A \ldots$ $B_1 - A_1 \ldots$)

2 samples.

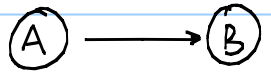
Average time / sample = $\frac{T_B + T_A}{2}$

Loop / Cycle : Loop bound = $\frac{\text{Total execution time on loop}}{\text{Total \# delays in loop}}$

For complete system: Iteration Period Bound = max loop bound.



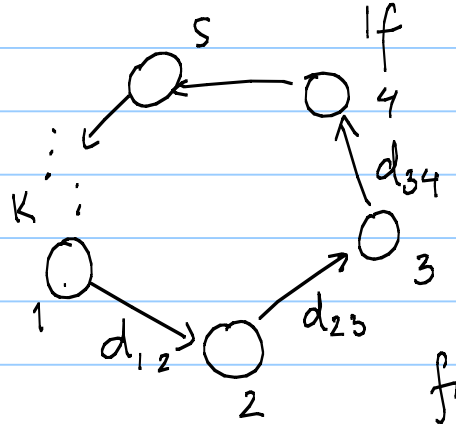
e_{ij} : $t_j \geq t_i + T_i$



$t_B \geq t_A + T_A$

starting time

execution time



If n_1 delays on e_{ij} : $t_j \geq t_i + T_i - n_1 T$

n_1 samples previously.

for a loop

$t_2 \geq t_1 - d_{12} T + T_1$

$t_3 \geq t_2 - d_{23} T + T_2$

$\vdots$

→ sample rate lower bound?

~~$t_2 + t_3 + \cdots$~~ $t_1 \geq$ ~~$t_1 + t_2 + \cdots$~~ $t_K - T(d_{12} + d_{23} + \cdots) + (T_1 + T_2 + \cdots T_K)$

$$T \cdot \sum d_{ij} \geq \sum T_i$$

$$T \geq \frac{\sum T_i}{\sum d_{ij}} \quad \text{for every loop in the system.}$$

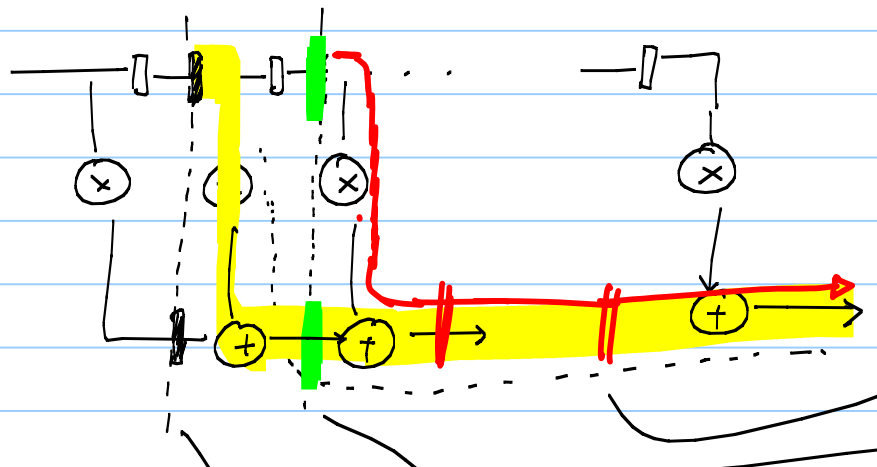
T = max over all loops of the loop bound.

$\longrightarrow$ minimum cycle mean (related problem)

- Iteration Period Bound (IPB)
  - depends on technology (T_i values)
  - depends on derivation of graph from eqns

No cycles in graph $\Rightarrow$ IPB = 0 $\Rightarrow$ Theoretical max sample processing = ∞

Pipelining and Parallelism

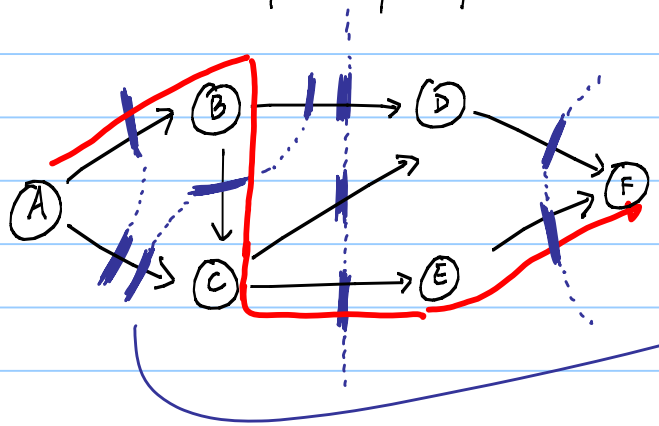


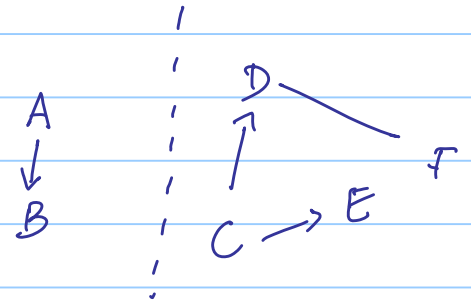
$CP = T_m + (N-1)\,T_a$.

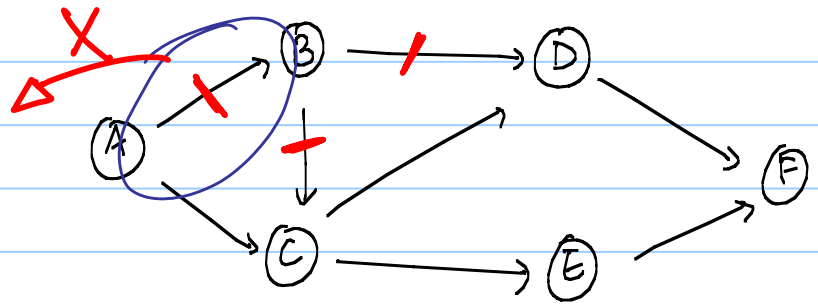
$CP = T_m + (N-1)\,T_a$.

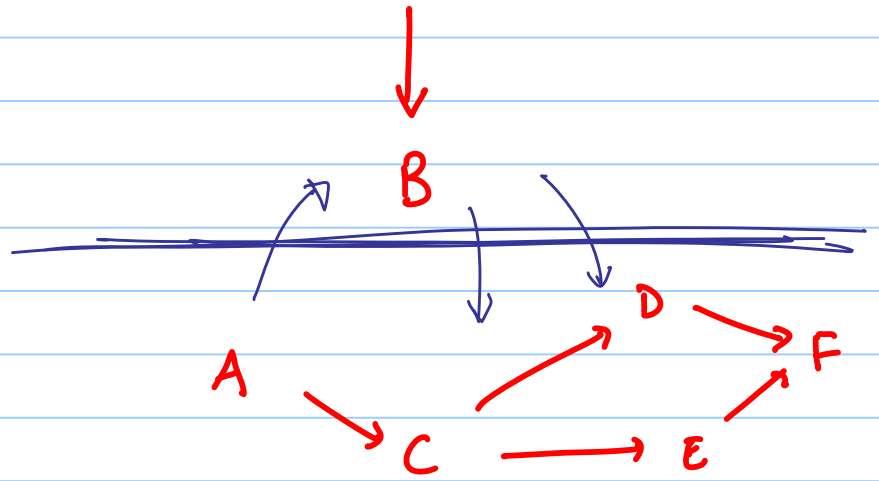
$CP = T_m + (N-2)\,T_a$.

Example feedforward cutset.

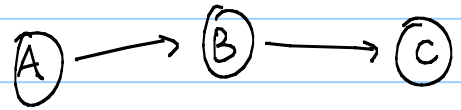








Not valid for pipelining.

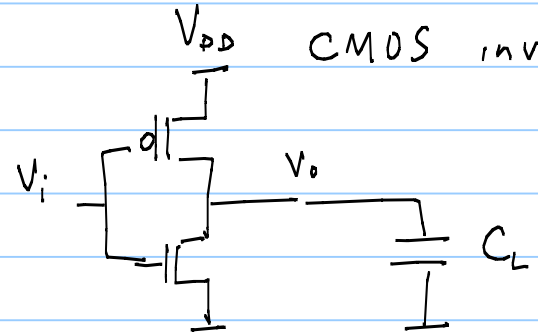


Copy ↓

Double amt. of hardware

<u>but</u>

also double throughput.



CMOS inverter

V_i low $\Rightarrow$ NMOS is off.
~ 0 PMOS is on

PMOS looks like current source

$$I = k_p (V_{GS} - V_T)^2$$

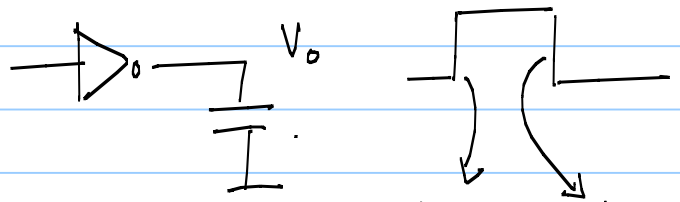
$V_{GS} \rightarrow V_{DD}$

C_L initially at 0 V
needs to be charged to V_{DD} when output is going low to high.

$$Q = C_L \cdot V_{DD}$$

$V_{DD} \uparrow$, prop delay $\downarrow$

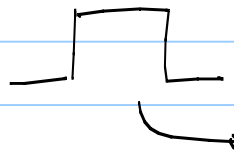
$$\text{Time} : \frac{C_L V_{DD}}{k (V_{DD} - V_T)^2}$$



$$E = \frac{1}{2} CV^2$$

$\frac{1}{2}CV^2$ dissipated in PMOS

$\frac{1}{2}CV^2$ dissipated in NMOS



CV^2 energy dissipated.

If output switching at frequency f

$$\text{Power} = \text{Energy/second} = \underline{CV^2 \cdot f} = C_L V_{DD}^2 \cdot f$$

Reduce power by reducing V_{DD} to optimum pt required for throughput.

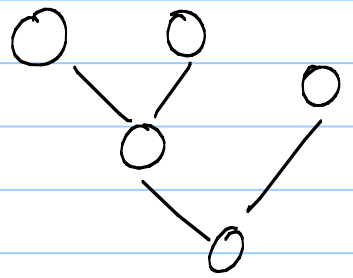
Pipelining : $C_L V_{DD}^2 f$

→ critical path reduced by pipelining, so reduce V_{DD} to get back to correct sample rate.

Parallelism : C_L (↑ increases) $V_{DD}^2 f$ (→ decreases)

↓

decrease → gives overall power saving.



→ map to 2 processors

Given a DFG, resources/processors,

- Allocation - allocate enough resources to solve the problem
- Binding - which operation happens on which processor
- Scheduling - when should each operation take place.

→ Architectural synthesis / High level synthesis.

NP-complete
- Non deterministic Polynomial-time
  NP

Most CAD algorithms, esp in DSP area are NP-hard.
→ Approximate, Randomized, Heuristic