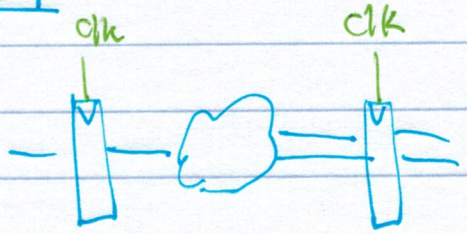
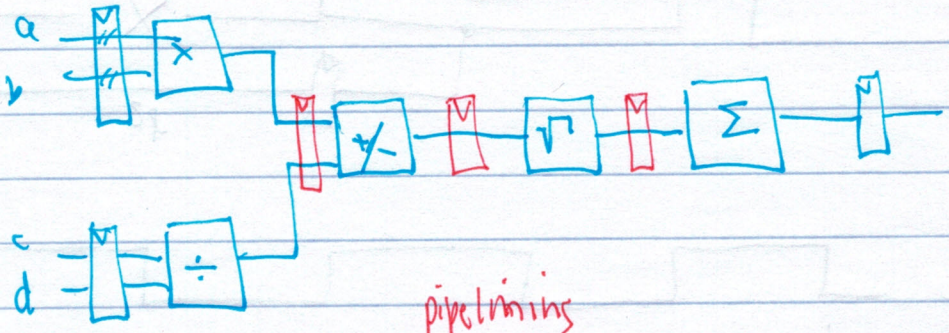
May 18, 2021

Pipelining



$$out = \sum \sqrt{(a \times b) \pm (c/d)}$$



pipelining

0000|0001|0000|0000

[-512, +511]

Ex: car factory
- engine
- paint
- install engine
- install interior

Break up logic (factory) into N pieces

Work on N pieces of data (cars) at a time

+ $f_{max} \simeq N \times$ higher

+ throughput $\simeq N \times$ higher

Not exactly true due to:

- t_{setup}
- $t_{clk\text{-}to\text{-}Q}$
- can't partition logic into equal pieces

– More area

– More energy per operation (almost certainly)

– Latency will increase $1\sim \rightarrow N\sim$

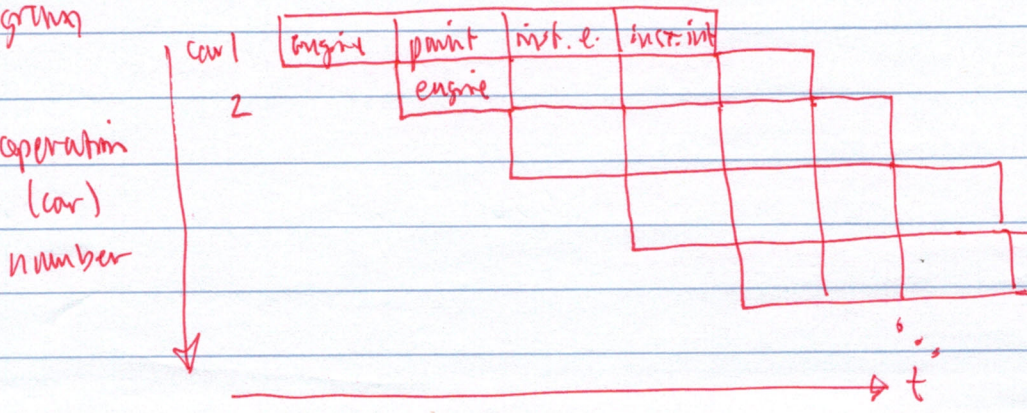
However clock frequency < ~N× higher

Ex: 1 GHz (1ns) → pipeline 4 stages → 3.2 GHz (0.31ns)

Latency 1.24ns now

– Control complexity
- Multiple ops in process at a time

Pipeline diagram



Ex:

clk-to-Q	150 ps
t_{setup}	100 ps
t_{hold}	125 ps
$t_{clk-skew}$	50 ps

Logic can be broken into pieces: 600 ps
500 ps
200
300
200
200 ps

① Unpipelined f_{max}?

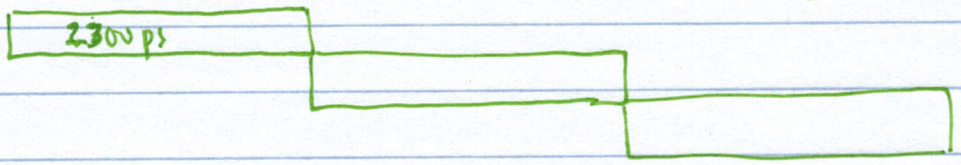
f_{max} = Min cycle time for clock

Min cycle time = 150 ps (clk-to-Q) + 2000 ps (logic) + 100 ps (setup) + 50 ps (skew)

= 2300 ps

f_{max} = 434 MHz

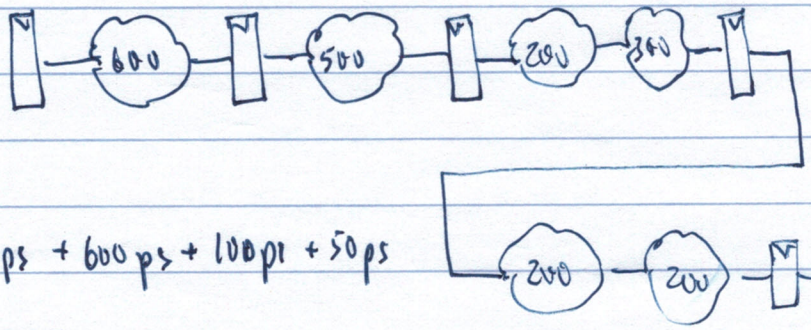
Overhead = 150 ps + 100 ps + 50 ps
= 300 ps



Throughput = 434 million ops/sec

Latency = 2300 ps

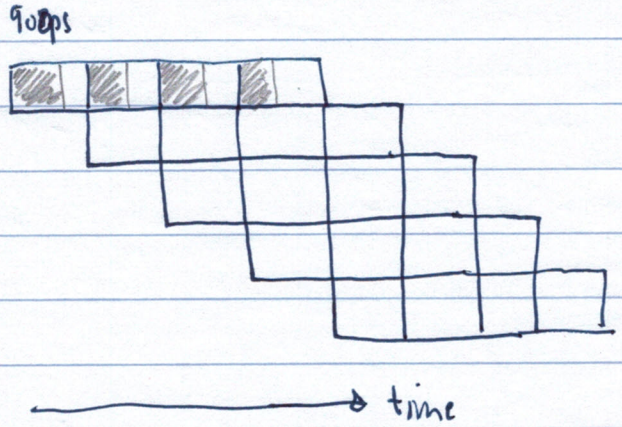
② Highest f_{max}



Min cycle time = 150ps + 600ps + 100ps + 50ps

= 900ps

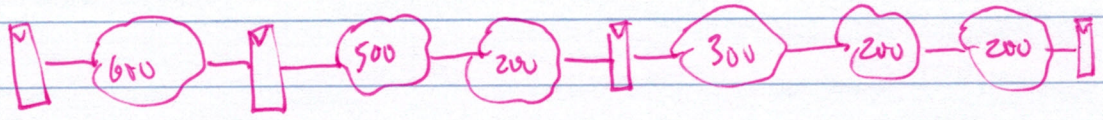
f_{max} = 1.11 GHz



Throughput = 1/900ps = 1111 million ops/sec

Latency = 4 × 900ps = 3600ps

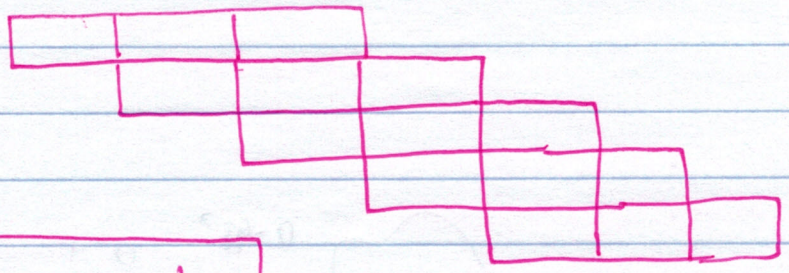
③



Min cycle time = 150ps + 700ps + 100ps + 50ps

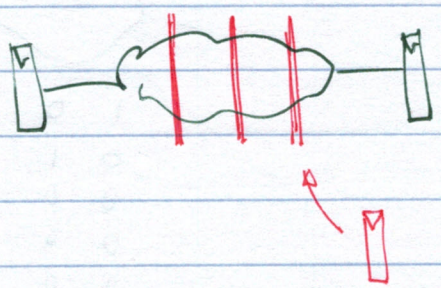
= 1000ps

f_{max} = 1.0 GHz



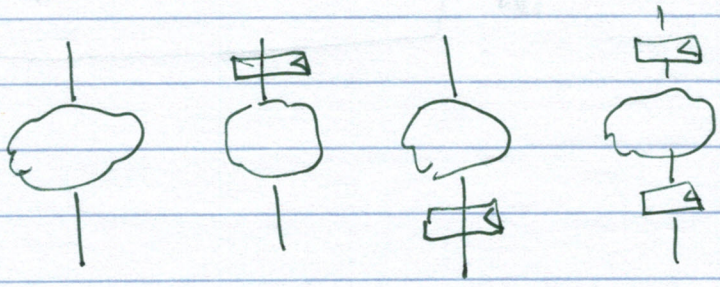
Throughput = 1000 million ops/sec

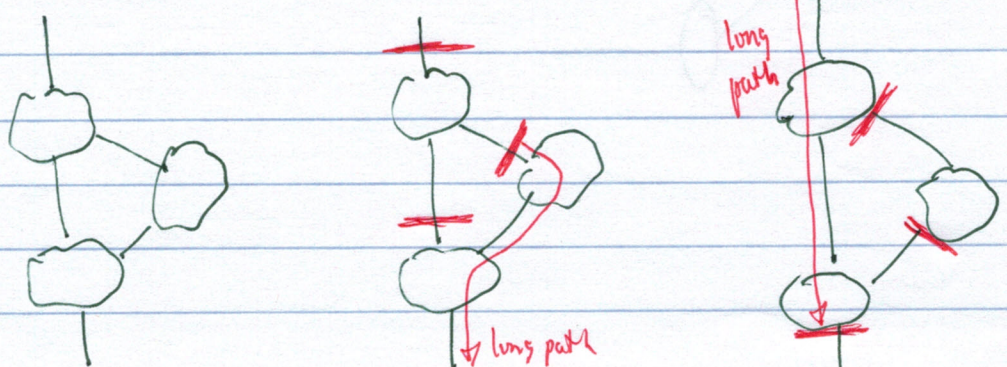
Latency = 3 × 1000 ps = 3000 ps



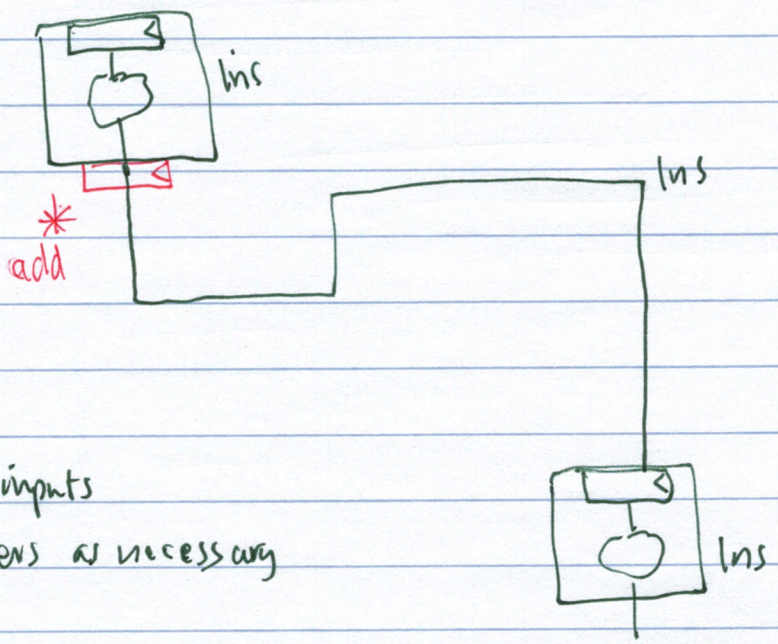
- Balanced delays in each stage ... ideally

- Ideally:
  - fewer stages
    - less latency
    - simpler control



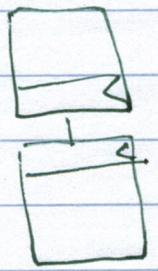


interconnect delays



Suggestion :-

- Always register all inputs
- Add output registers as necessary



Memory

I. Single-bit

* II. Array

Used in gen. purp. processors

1) Instructions
2) Data

Special-purp. proc.

1) Buffering
2) Fixed constants
3) Often relatively small (8-64-1024 words common)

Key design goal : Density Area

- Datapaths
- Control
- Memory