Pipelining

Adapted from Rajeev Subramanium's Spring '16 CS6810 course (University of Utah)

A correction has been done in slide No.8, which was incorrect in the video, i.e., the version shown in class.

4 Feb 2019

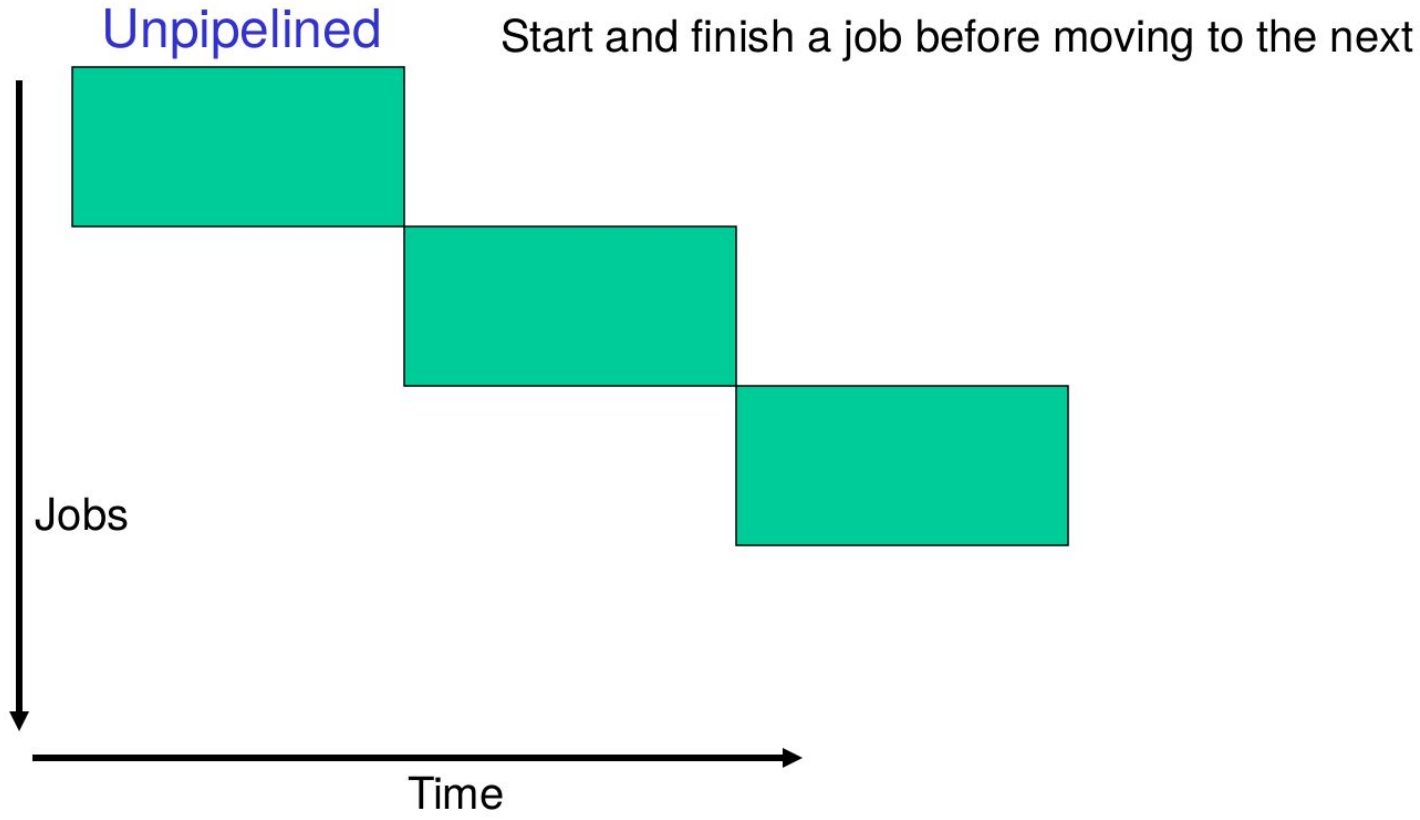
Aftab Hussain
University of California,
Irvine

A data processing technique

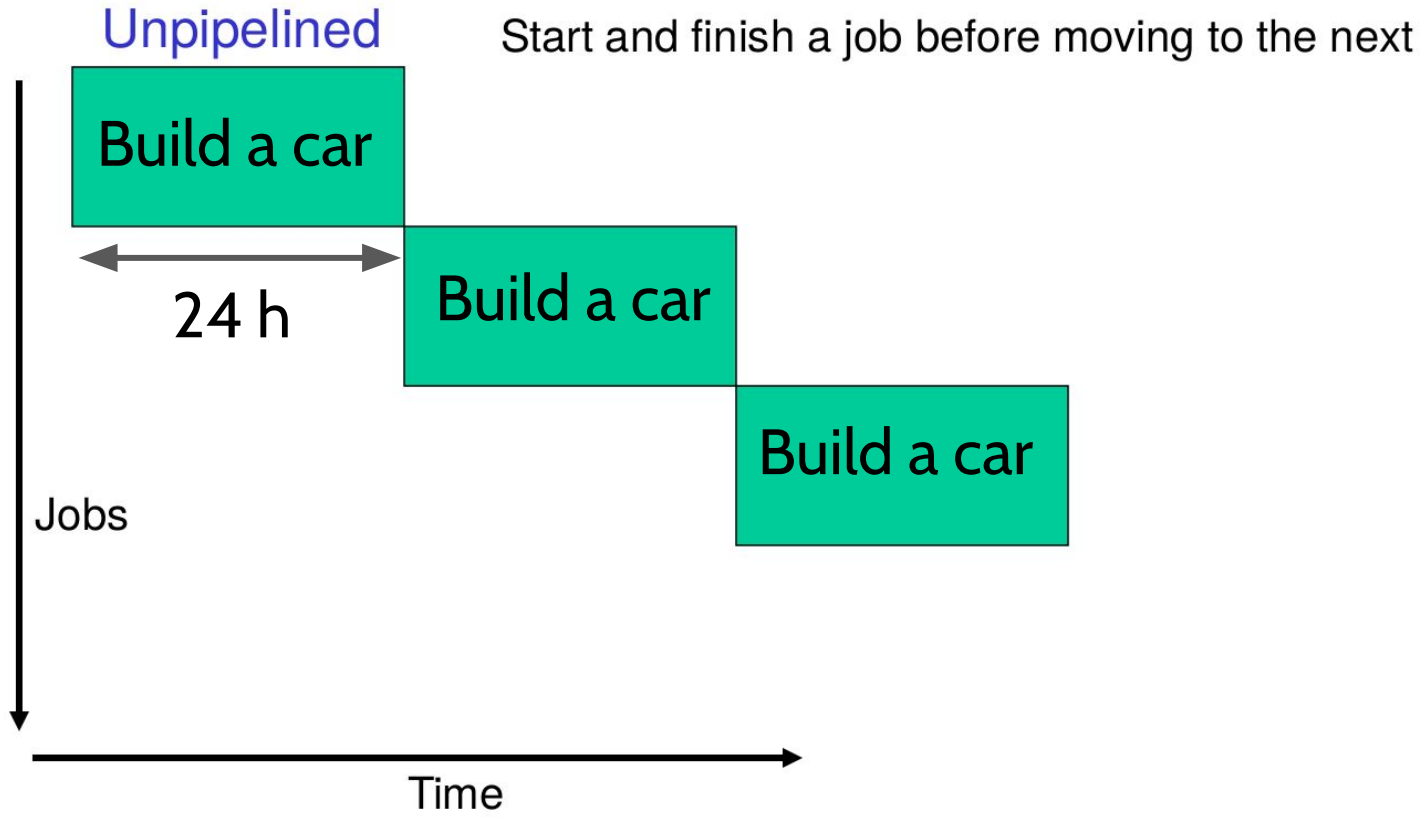
where the data processing elements
are connected in a series

and some of those elements are executed in parallel using time slicing.

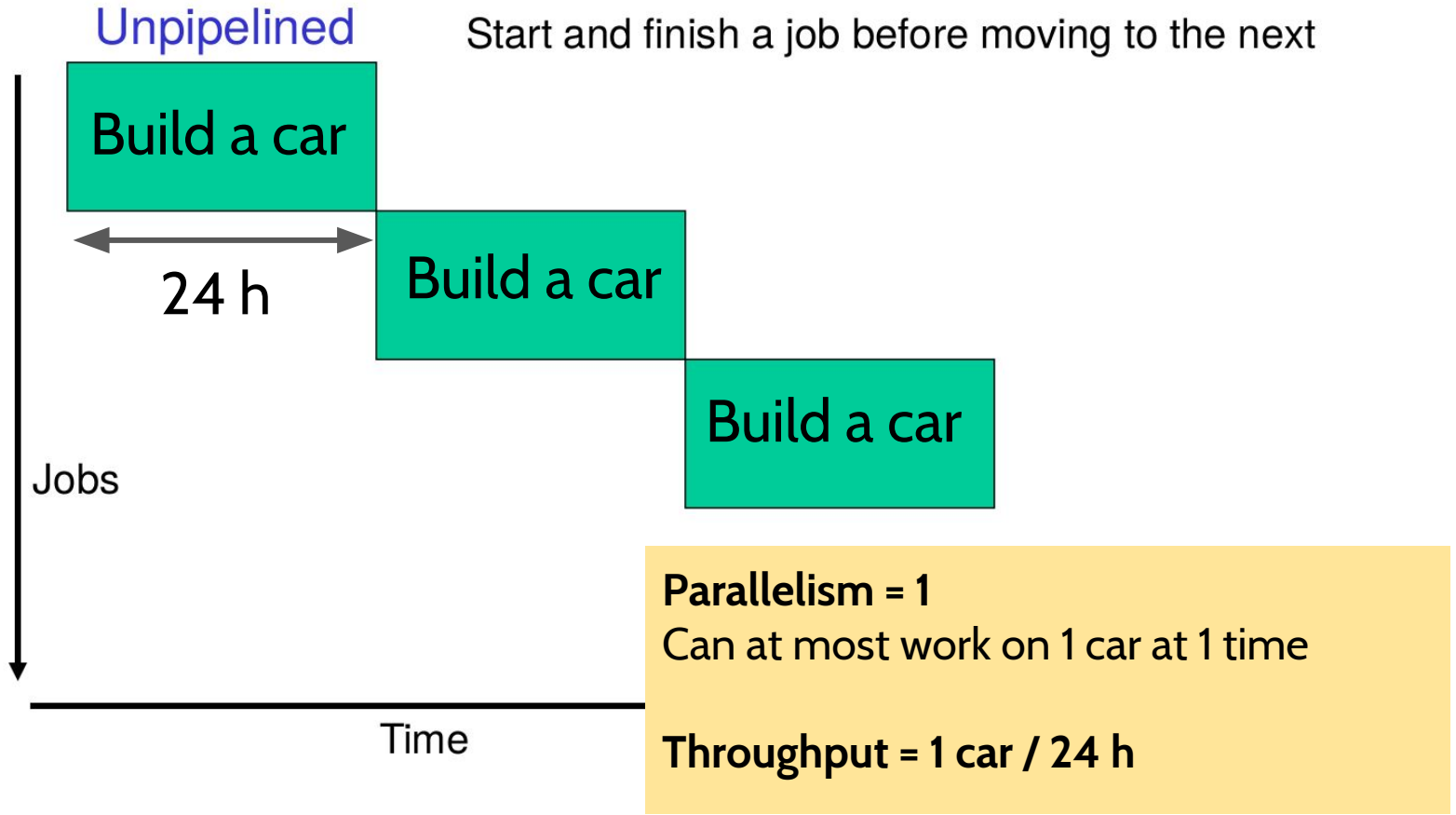
https://en.wikipedia.org/wiki/Pipeline_(computing)



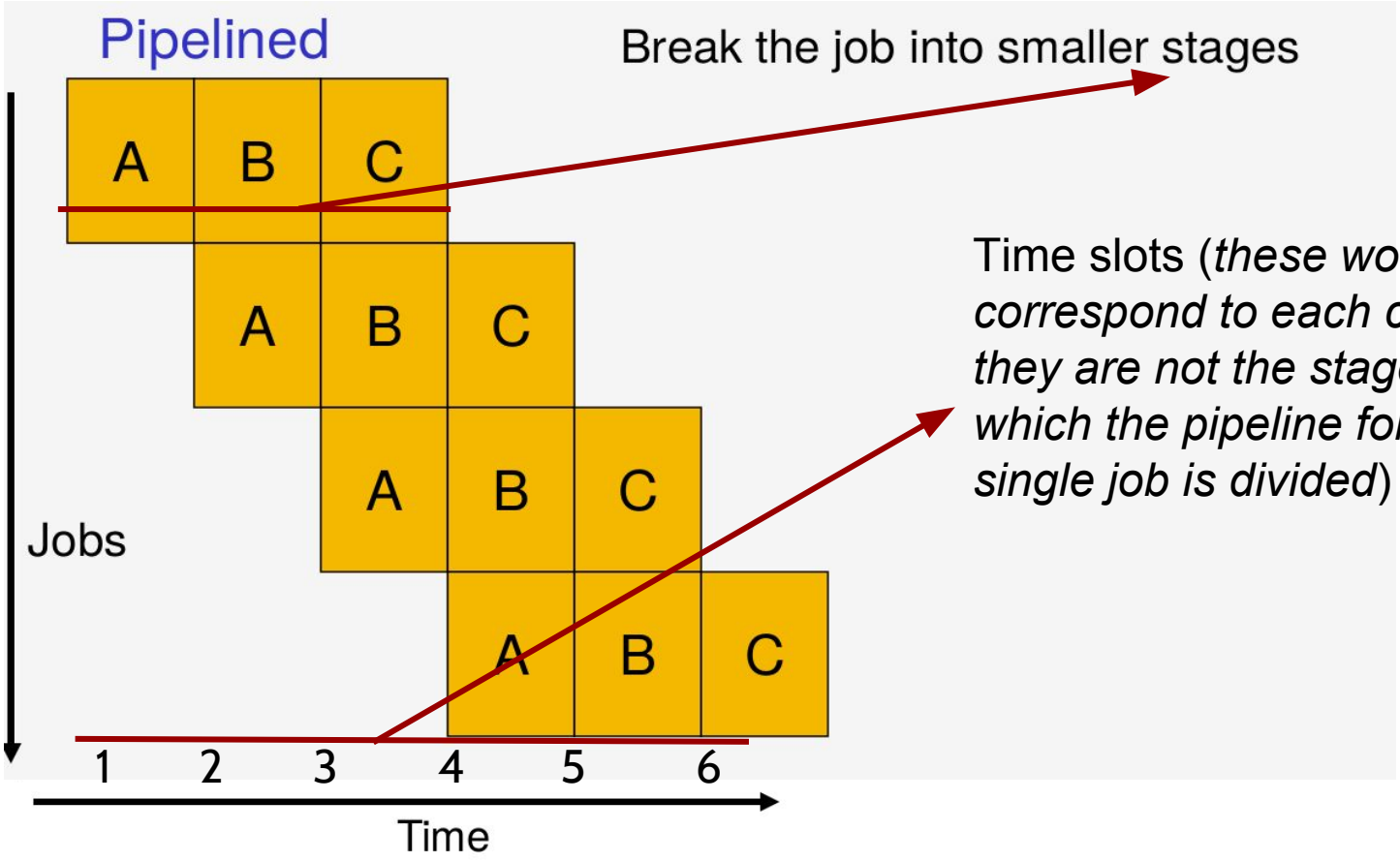
http://www.cs.utah.edu/~rajeev/cs6810/pres/12-6810-03.pdf

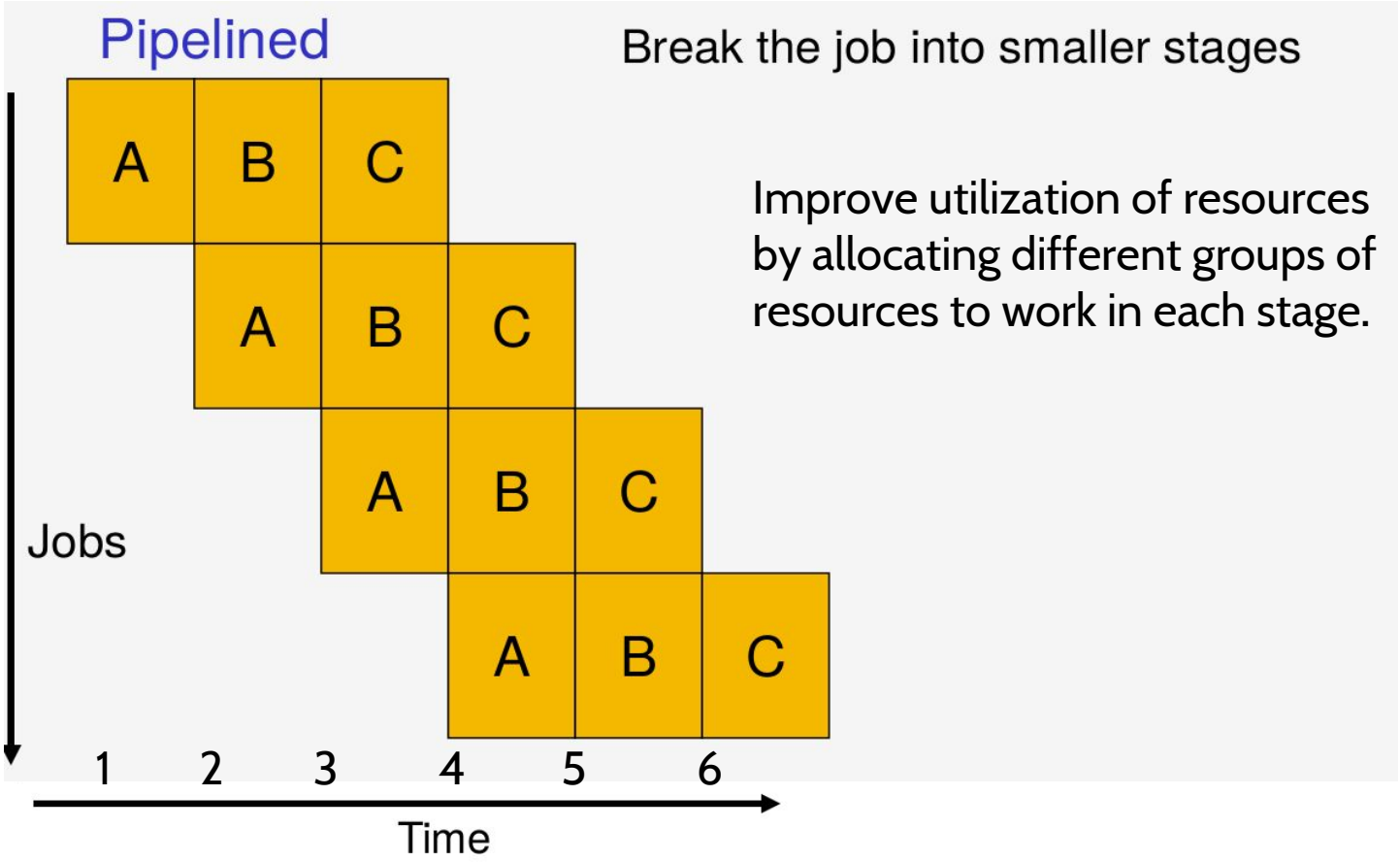


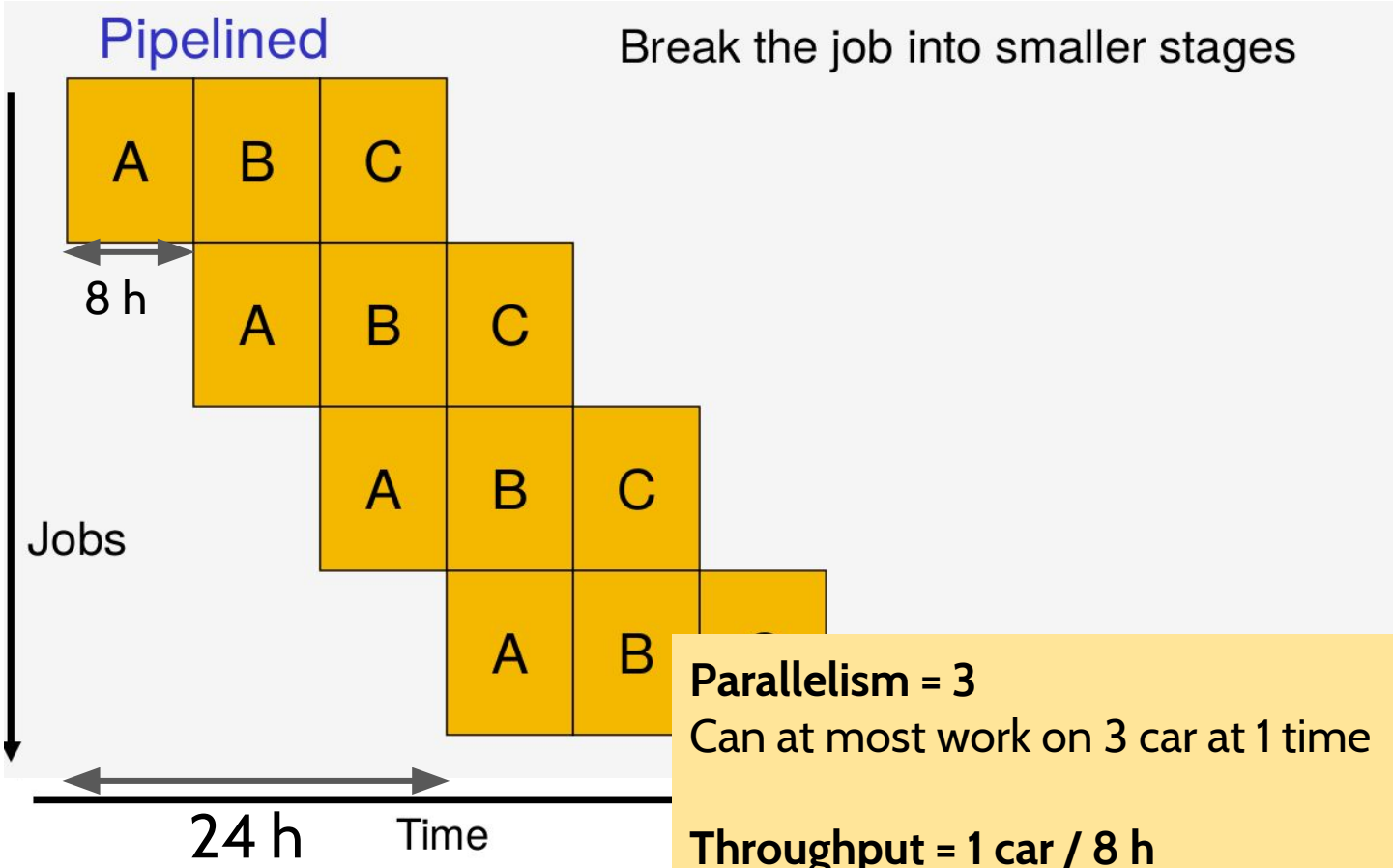
http://www.cs.utah.edu/~rajeev/cs6810/pres/12-6810-03.pdf



http://www.cs.utah.edu/~rajeev/cs6810/pres/12-6810-03.pdf


Pipelined
Break the job into smaller stages
A
B
C
A
B
C
A
B
C
A
B
C
Jobs
1
2
3
4
5
6
Time
Time slots (these would correspond to each cycle - they are not the stages into which the pipeline for a single job is divided)

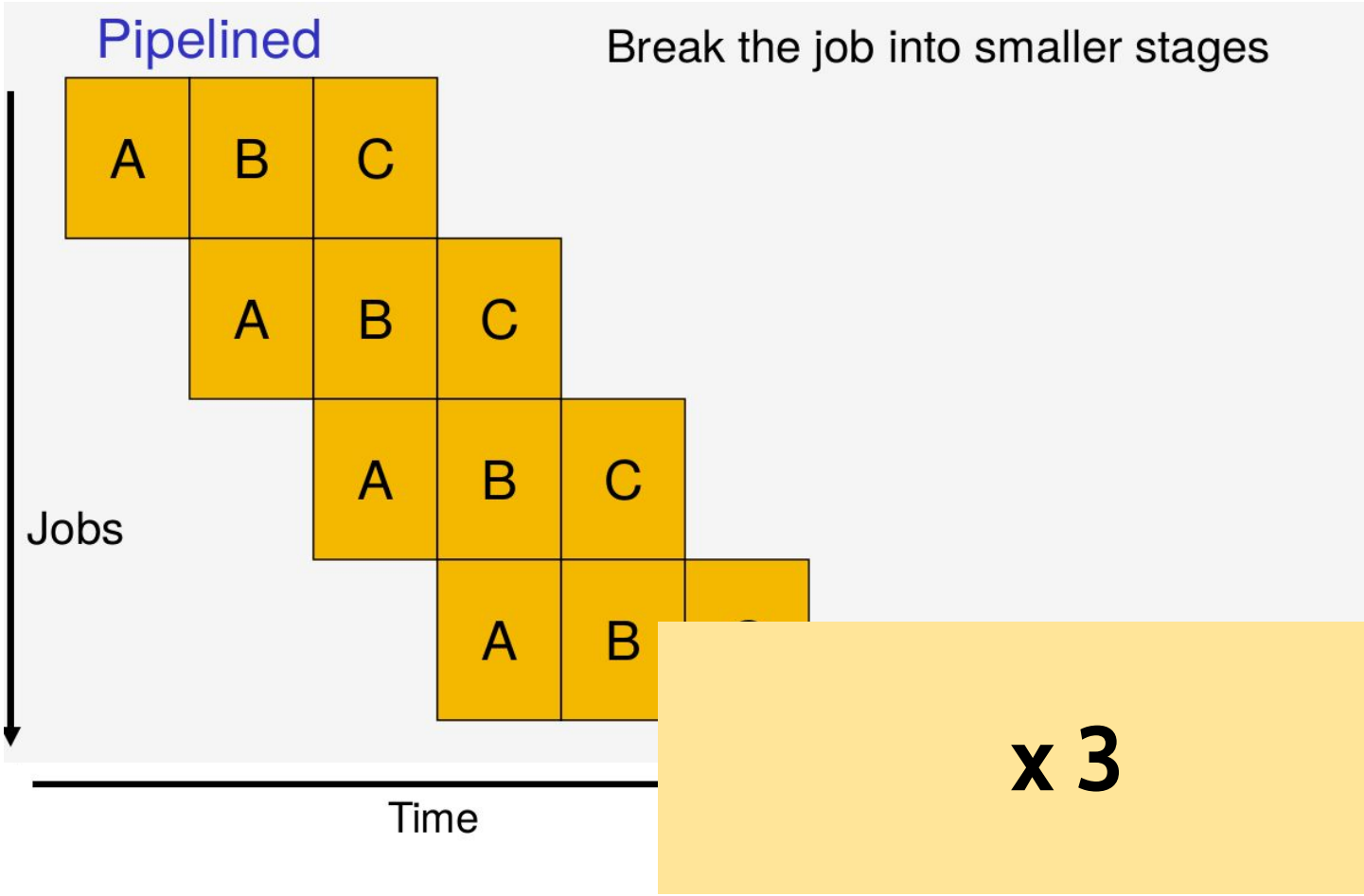

Pipelined
Break the job into smaller stages
A
B
C
A
B
C
A
B
C
A
B
C
Jobs
1
2
3
4
5
6
Time
Improve utilization of resources by allocating different groups of resources to work in each stage.



Parallelism = 3
Can at most work on 3 car at 1 time

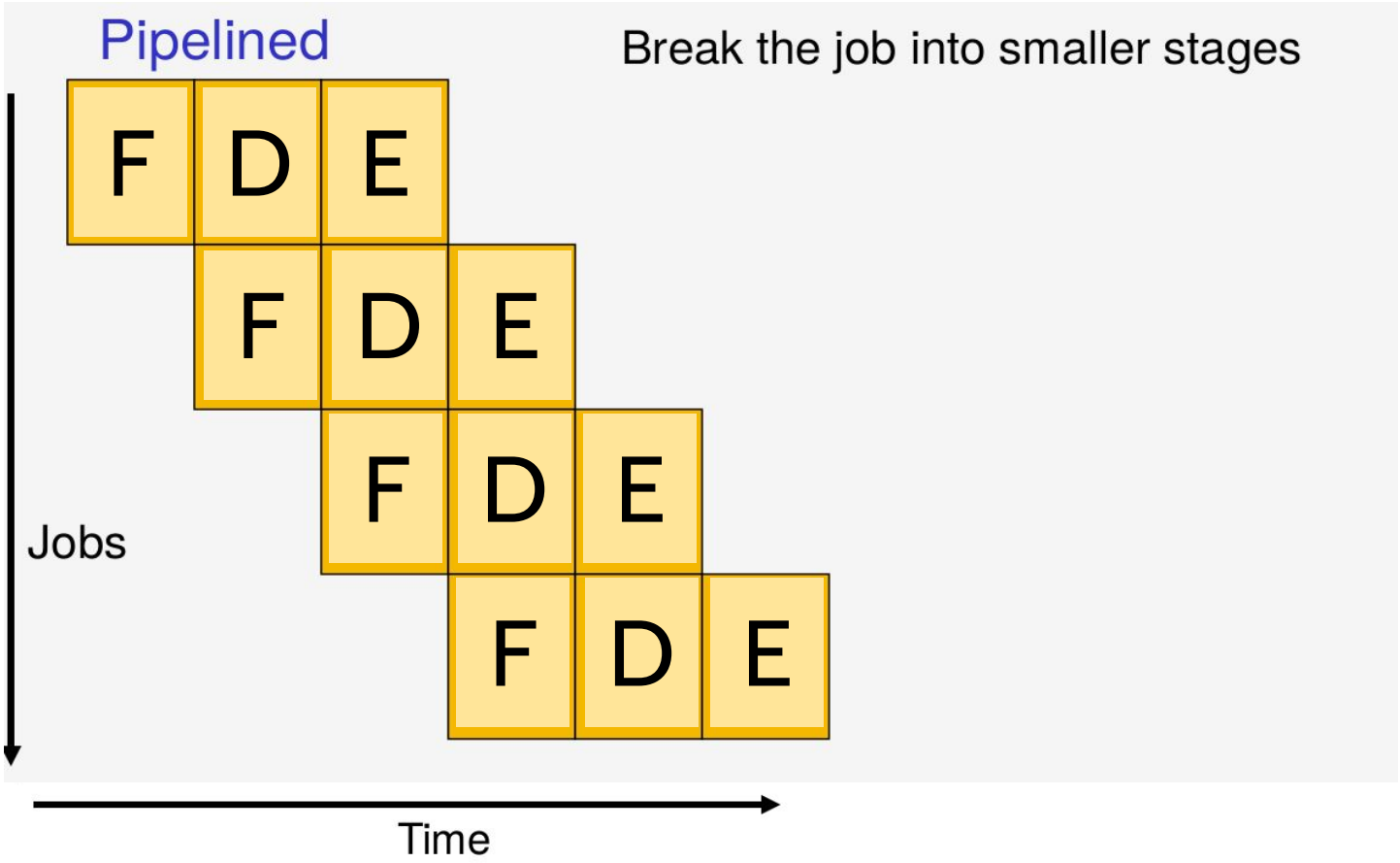
Throughput = 1 car / 8 h

http://www.cs.utah.edu/~rajeev/cs6810/pres/12-6810-03.pdf

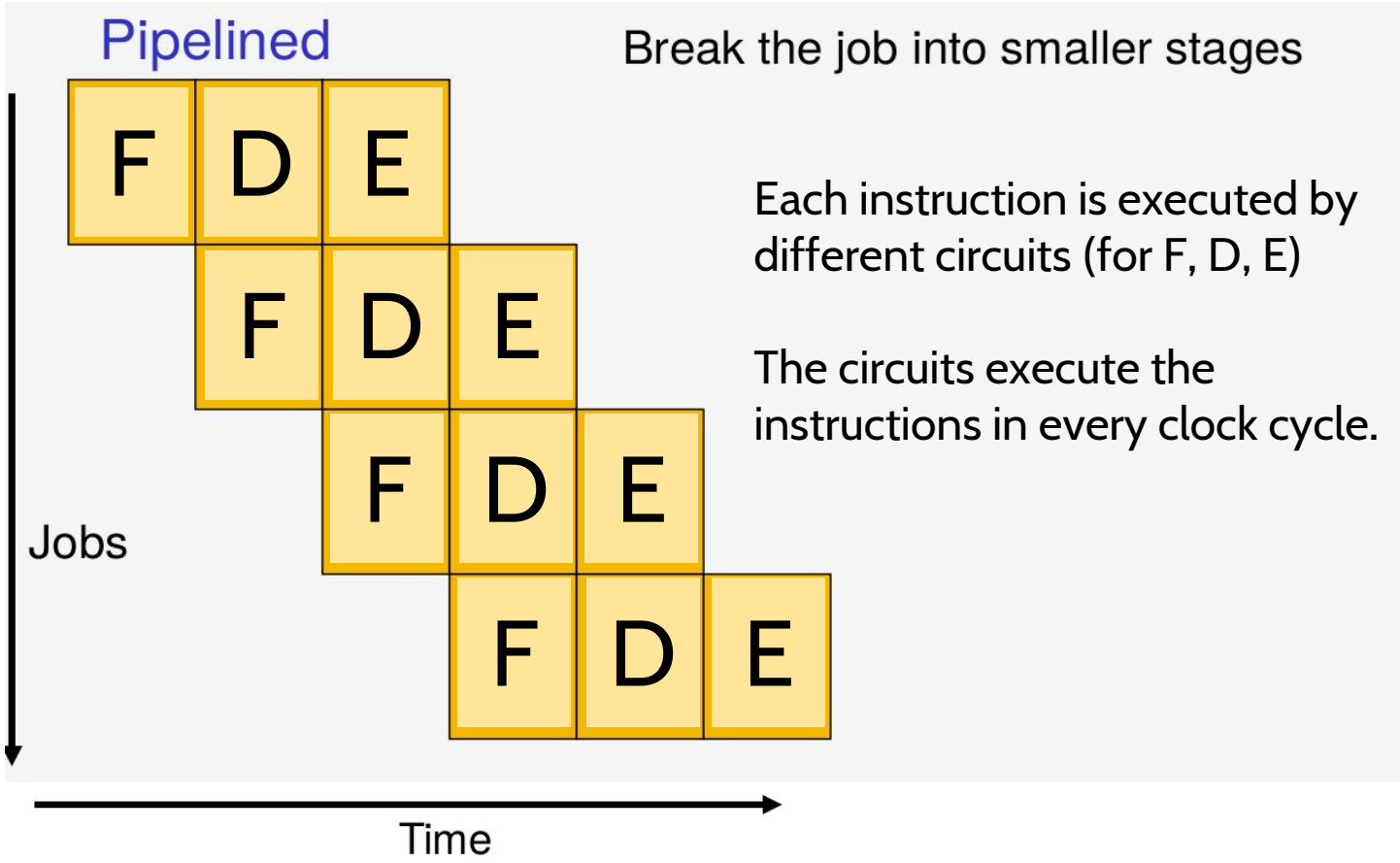


http://www.cs.utah.edu/~rajeev/cs6810/pres/12-6810-03.pdf

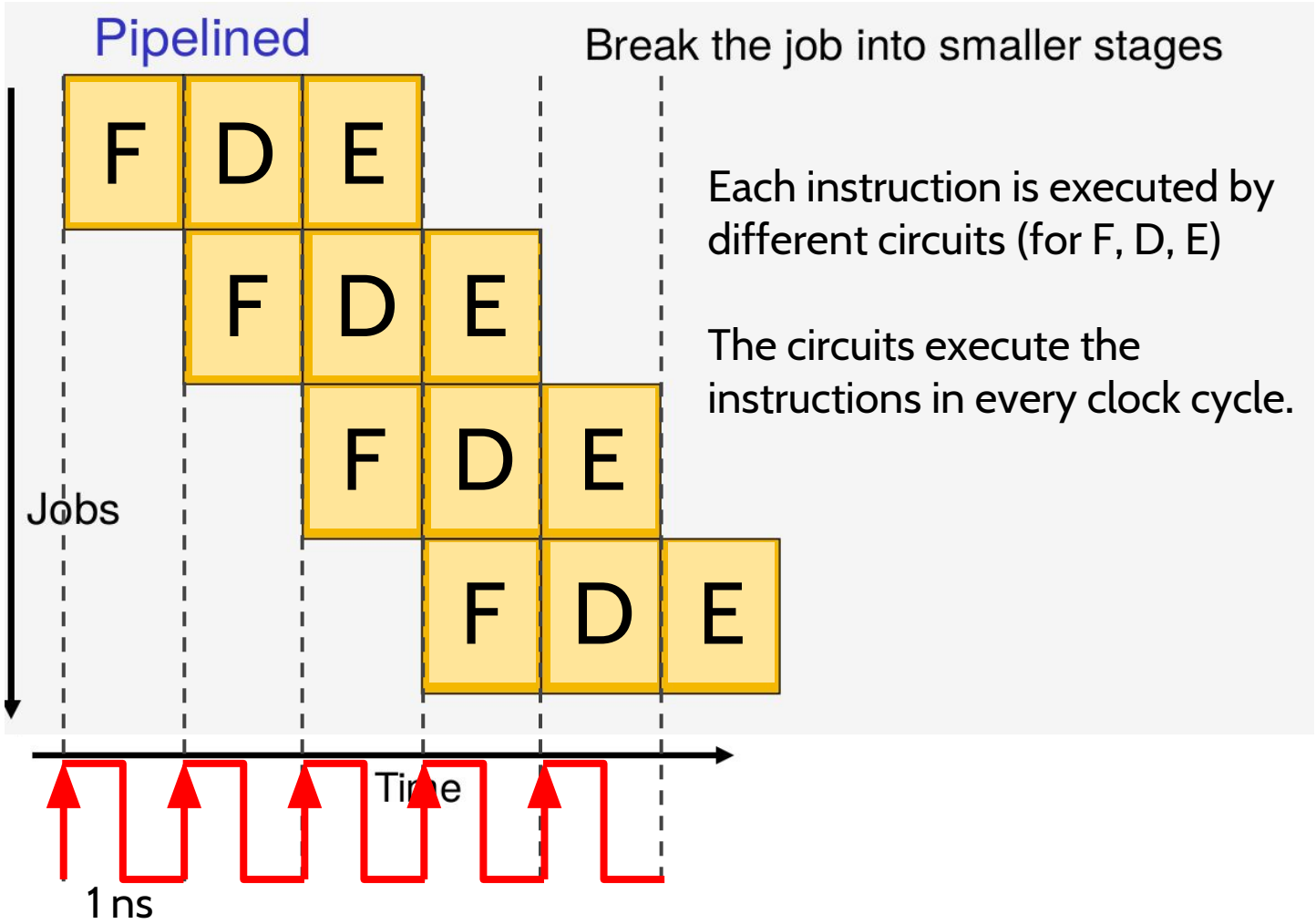
Instructions



http://www.cs.utah.edu/~rajeev/cs6810/pres/12-6810-03.pdf



http://www.cs.utah.edu/~rajeev/cs6810/pres/12-6810-03.pdf

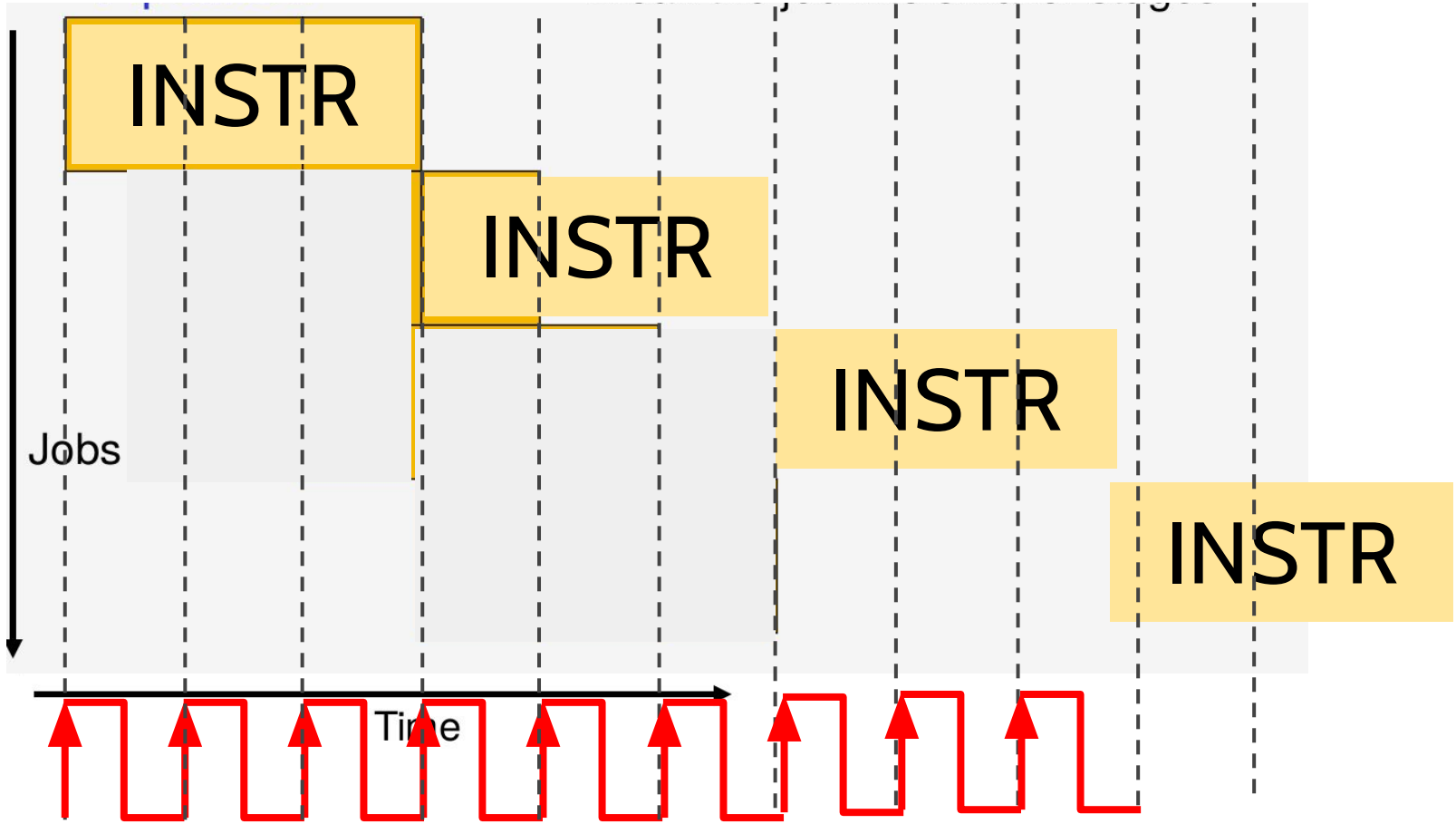


Each instruction is executed by different circuits (for F, D, E)

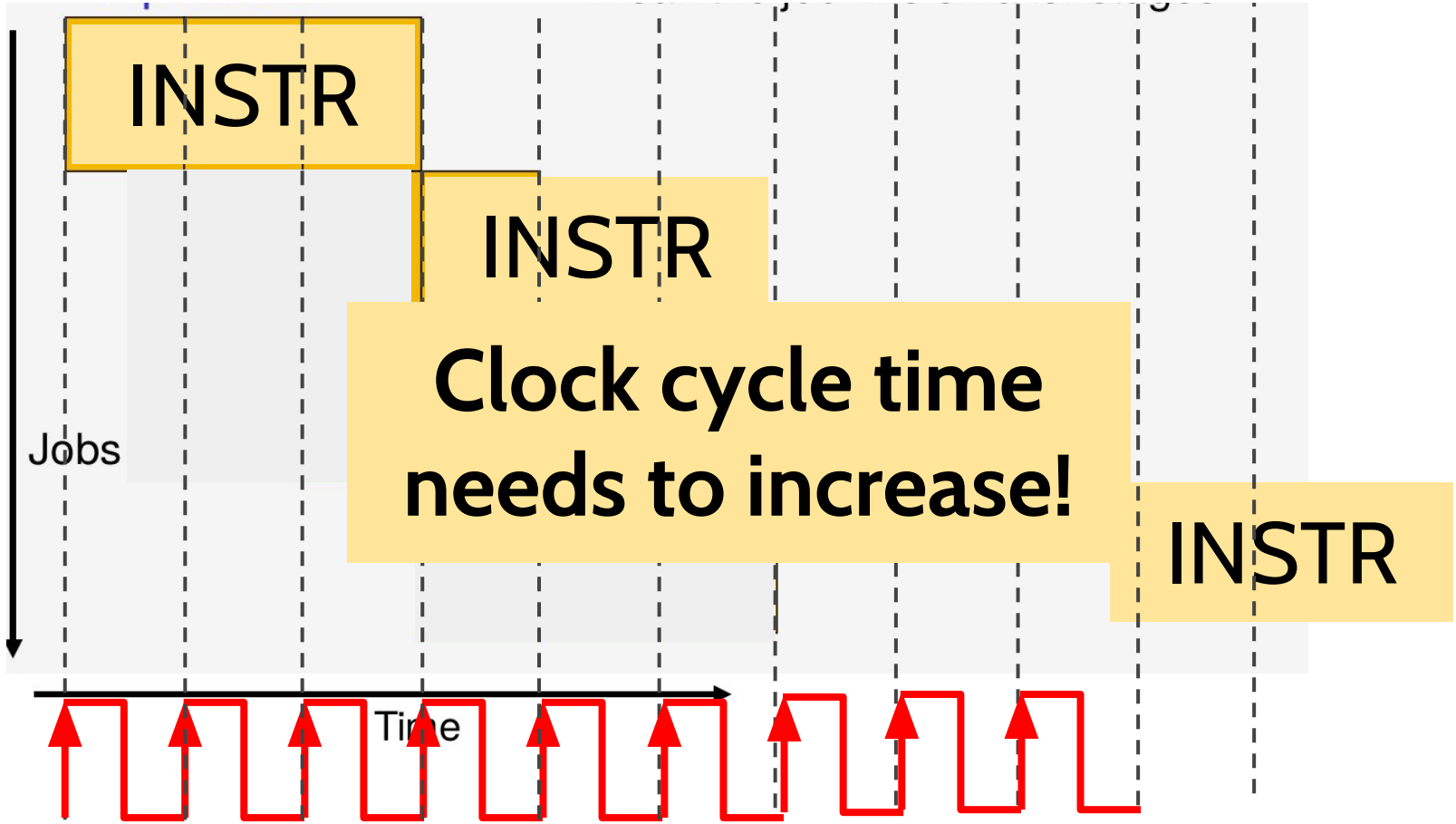
The circuits execute the instructions in every clock cycle.

http://www.cs.utah.edu/~rajeev/cs6810/pres/12-6810-03.pdf

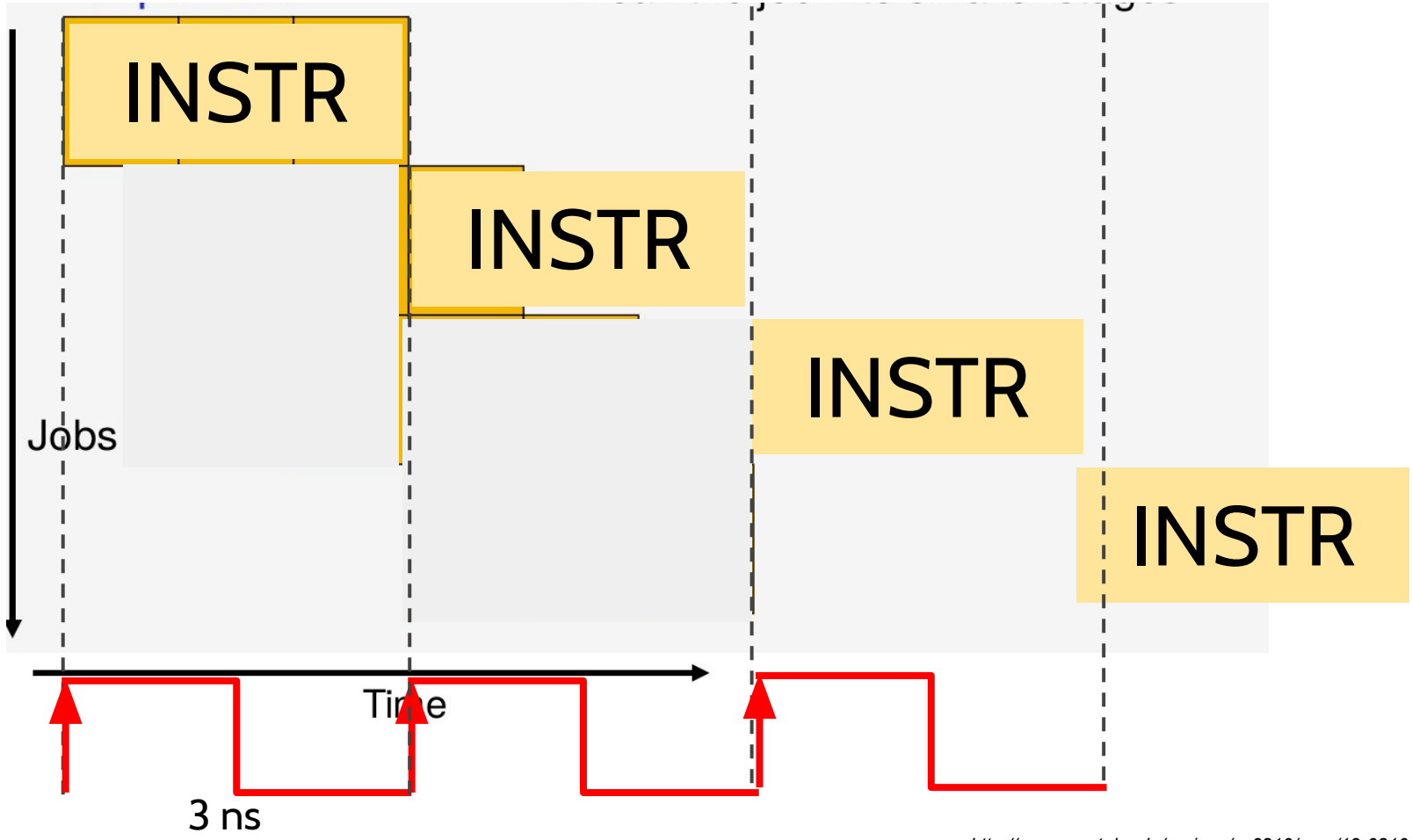
Let’s see how instruction execution happens in unpipelined mode



http://www.cs.utah.edu/~rajeev/cs6810/pres/12-6810-03.pdf



http://www.cs.utah.edu/~rajeev/cs6810/pres/12-6810-03.pdf



http://www.cs.utah.edu/~rajeev/cs6810/pres/12-6810-03.pdf

Overhead in pipelining - latches

Overhead in pipelining - latches

Need for latches - [White-Board]

At every rising clock edge, a latch

+ samples whatever value is on its I/P

+ stores it in its output

+ retains it until the next rising clock edge

Pipelining Equations

- Unpipelined: time to execute one instruction = $T + T_{ovh}$
- For an N-stage pipeline, time per stage = $T/N + T_{ovh}$
- Total time per instruction = $N\ (T/N + T_{ovh}) = T + N\ T_{ovh}$
- Clock cycle time = $T/N + T_{ovh}$
- Clock speed = $1 / (T/N + T_{ovh})$
- Ideal speedup = $(T + T_{ovh}) / (T/N + T_{ovh})$
- Cycles to complete one instruction = N
- Average CPI (cycles per instr) = 1

Problems 1, 2 on Rajeev's Slides (Slide nos. 5 - 8)

Thank you