High Level Synthesis

Power Issues

Virendra Singh

Computer Architecture and Dependable Systems Lab

Department of Electrical Engineering

Indian Institute of Technology Bombay

http://www.ee.iitb.ac.in/~viren/

E-mail: viren@ee.iitb.ac.in

EE-677: Foundations of VLSI CAD

Lecture 11 on 26 August 2021

CADSL

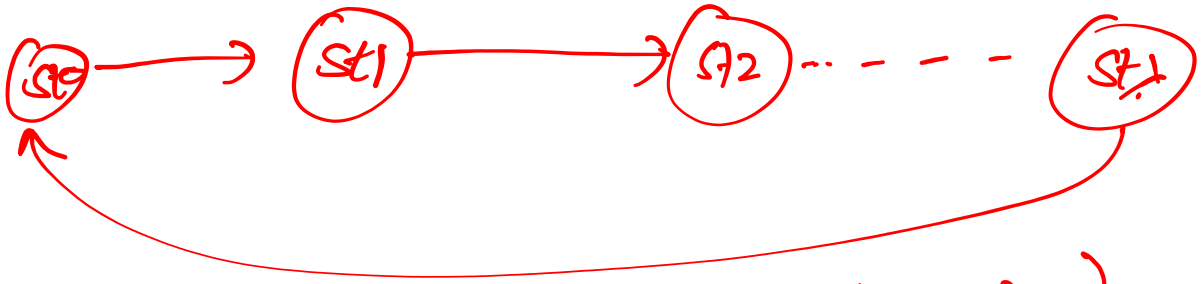
HLS

Datapath
- FU ✓
- Storage ✓
- Steering Logic

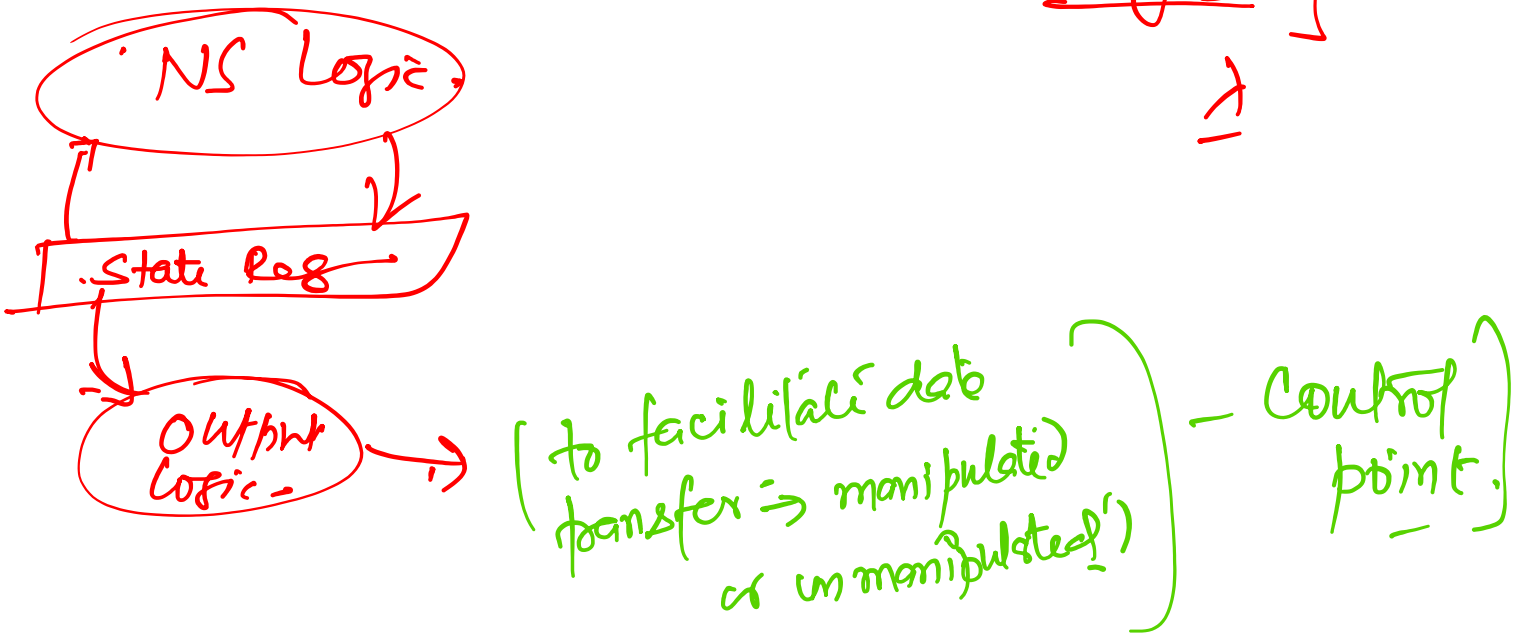
DFG → Sequencing Graph
- Operations & dependencies
- FU
  - Place in time & space.
    - time → scheduling
    - space → Binding.

- liveness range. (life time) analysis
  - Register allocation / Binding.
- Steering Logic.

CONTROLLER → FSM ⇒ #State = λ State transition → St0 → St1 - - - → Stλ



$\log_2 \lambda$ / λ



OPTIMISATION

PPTAS2

1. Area (Cost) ✓

2. Performance. ✓

2. Area + (1-2) Perf...

POWER.

High Level Synthesis

Objective

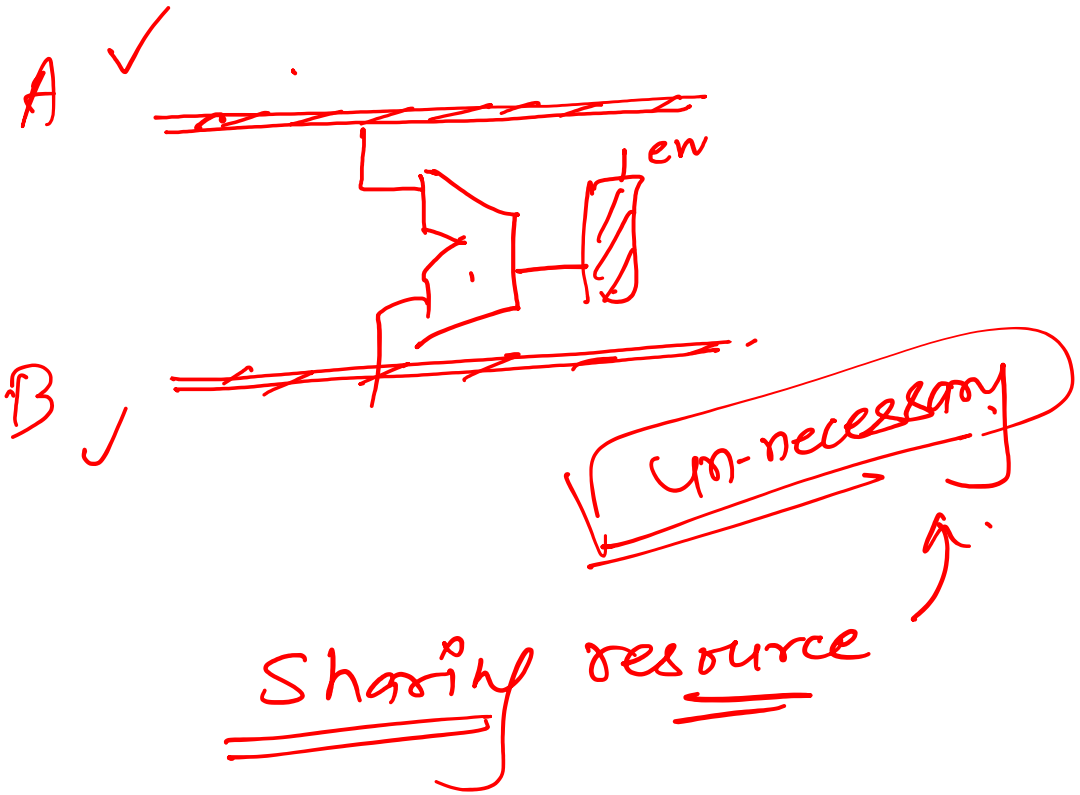
- Area
- Performance
- Power
- Reliability

Power

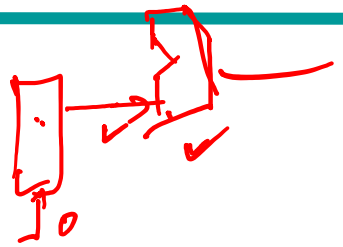
- Dynamic Power
- Static Power
- Transitions

- Large fraction of transitions incurred during the circuit operation are unnecessary

➢ Suppressing or eliminating unnecessary transitions


A
B
Broadcast.
BUS
un-necessary
Sharing resource

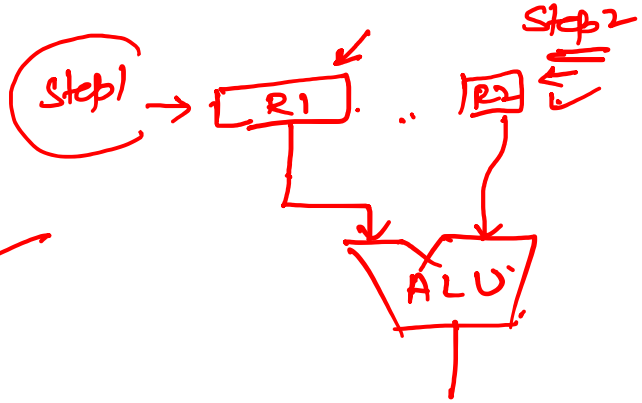
Power Reduction Techniques



Techniques

- Clock gating

- Partitioning of circuits



- Scheduling – to maximize idle time
- Maximize the sleep time of storage elements

Power Reduction Techniques

During the control steps in which functional unit is utilized to perform some operation is said to be active

During other control steps, the functional unit is said to be idle

The manner in which register sharing is performed can significantly affect the unnecessary power dissipation (spurious switching) in functional units during their idle cycle

Effect of Register Sharing

Ganesh et al [TVLSI, 1999]

Architectural Model

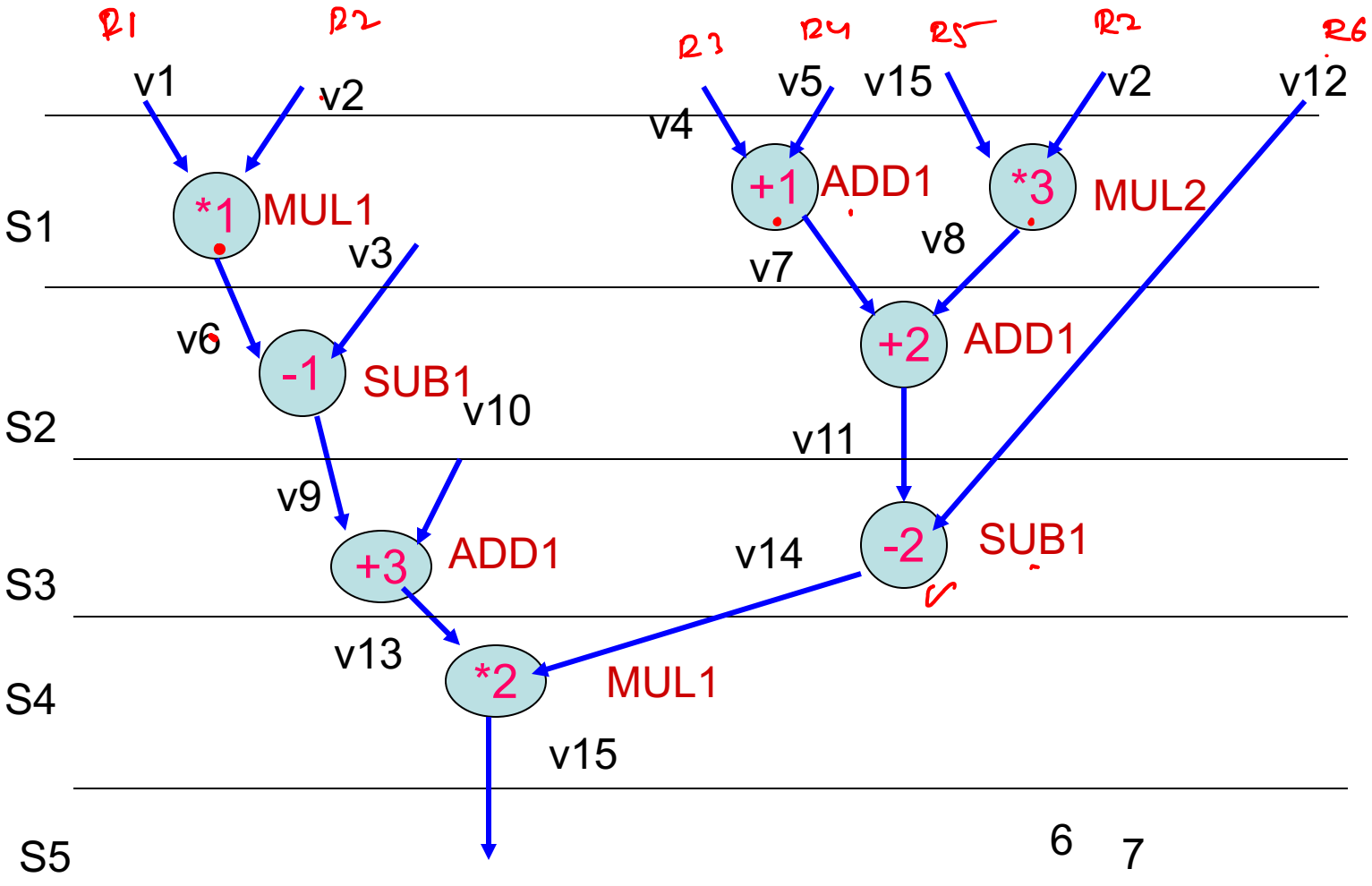
Register Allocation and variable assignment

Effect on switching activity

Spurious switching activities can sometimes be eliminated without increasing the number of registers in synthesized circuit

Scheduled DFG

left, edge



Variable Assignment

left edge algo.

Register	Assignment1
R1	V1, V7, V11, V13
R2	V2, V8, V10, V14
R3	V3, V5, V9
R4	V4, V6
R5	V12
R6	V15

Switching Activity

S1	V1	*1	V2 V15	*3	V2	X	V12 V4	+1	V5
S2	V7	X	V8	X	V8 V6	-1	V3 V7	+2	V8
S3	V11	X	V10	X	V10 V11	-2	V12 V9	+3	V10
S4	V13	*2	V14	X	V14 V13	X		X	V14
S5									
		MUL1		mul 2		Sub1		Add 1	

Alternate Variable Assignment

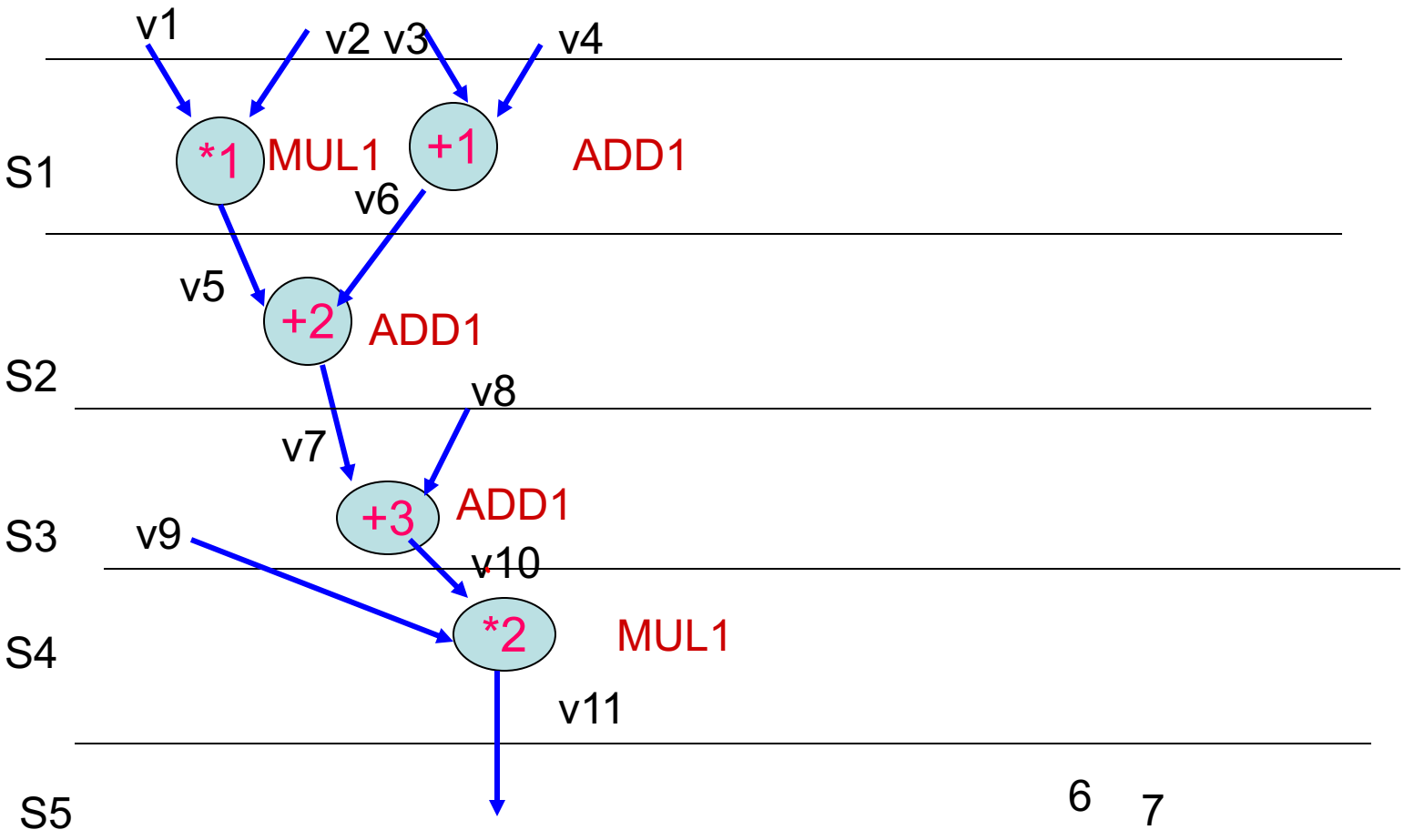
Register	Assignment2
R1	V1, V13
R2	V2
R3	V4, V8, V10
R4	V5, V7, V9
R5	V12
R6	V3
R7	V6, V11
R8	V14
R9	V15

3 additional Registers.
Overhead.

Switching Activity

S1	V1	*1	V2 V15	*3	V2	X	V12 V4	+1	V5
S2					V6	-1	V3 V7	+2	V8
S3					V11	-2	V12 V9	+3	V10
S4	V13	*2	V14						
S5									
		MUL 1		mu l2		Add1		Add 2	

Scheduled DFG



Variable Assignments

left edge also.

No. extra register is needed.

Register	Assignment1	Assignment
R1	V1, V5, V7, V9	V1, V9
R2	V2, V6, V8, V10	V2, V10
R3	V3	V3, V5, V7
R4	V4	V4, V6, V8

only necessary activity.

IKV5E5NKI

QZHTVY

Quiz 1. password.

Power Management Technique

Perfect Power Management{

- For each functional unit *fu*{
  - for each operation mapped to *fu* (*op*){
    - if op is the last operation born in fu return TRUE;
    - for each input variable v_{ini} of operation *op*{
      - tbirthLnext <- birth time of left input of the operation succeeding op on fu
      - tbirthRnext <- birth time of right input of the operation succeeding op on fu
      - SET_DEATH_TIME (vin_i, max (tbirthLnext, tbirthRnext);

Power Reduction Techniques

- if (LIFETIME (vin_i) conflicts with lifetime of other variable mapped to the same register) return FALSE;
- Set multiplexers at *fu's* input to select vin_i

  until max ($tbirthL_{next}$, $tbirthR_{next}$);

}

}}

Thank You