VLSI DESIGN FLOW: RTL TO GDS

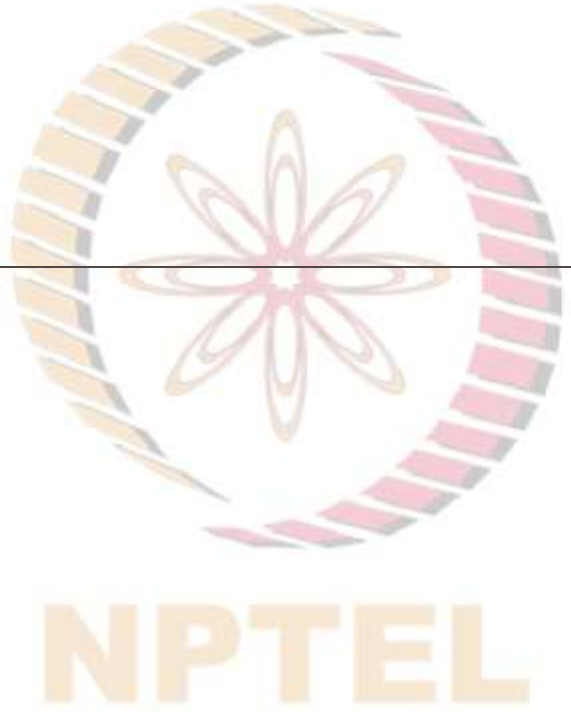
Lecture 4
Overview of VLSI Design Flow: II

Sneh Saurabh
Electronics and Communications Engineering
IIIT Delhi

Lecture Plan

Pre-RTL Methodologies

- Functional description to RTL
- Reusing RTL
- Behavior Synthesis



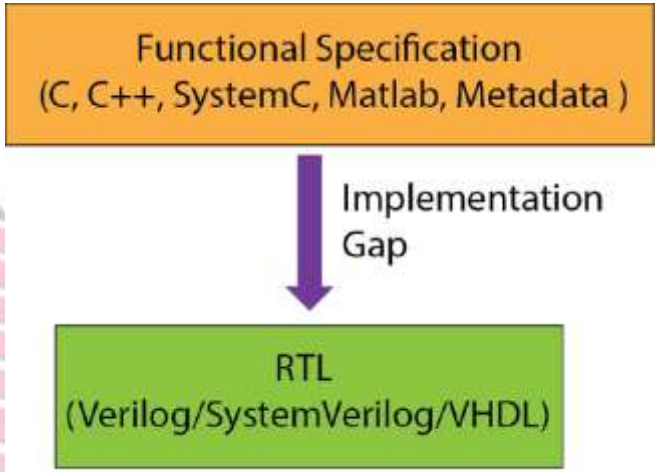
Pre-RTL Methodologies

Functional Specification to RTL

Functional Specification

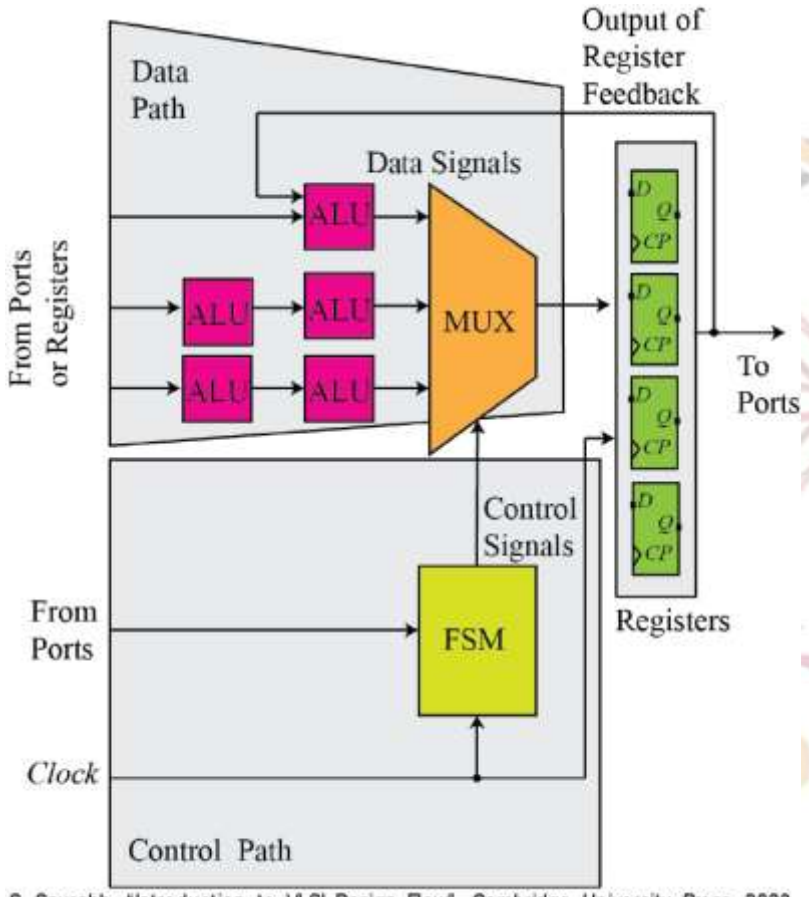
- Functional Specification can be made at a higher level of abstraction
- Opens up *implementation gap*

- Need to convert to RTL
  - Describes data flow from register to register at various time instants or clock cycle
  - Carries timing information



S. Saurabh, "Introduction to VLSI Design Flow". *Cambridge University Press*, 2023.

RTL: General Structure



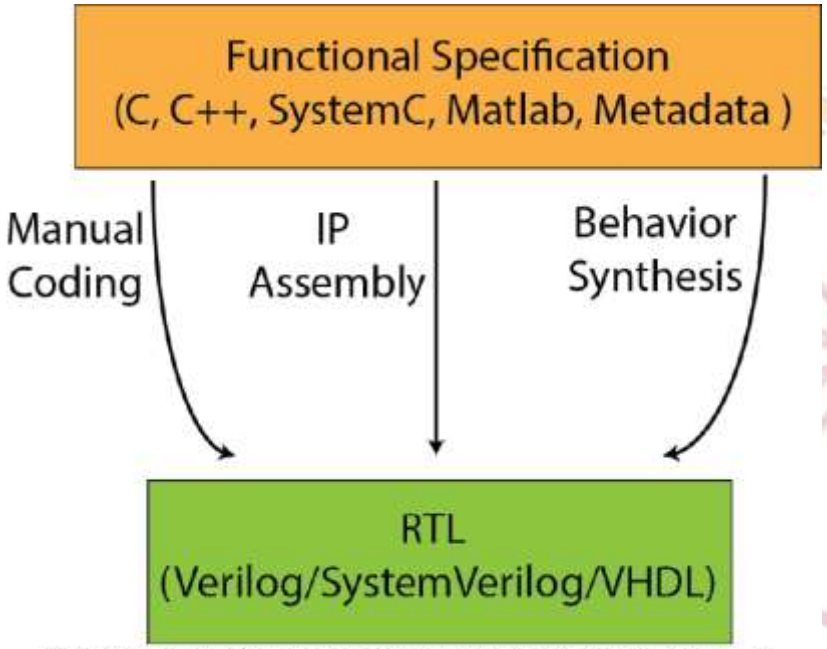
S. Saurabh, "Introduction to VLSI Design Flow". *Cambridge University Press*, 2023.

RTL: Register Transfer Level

- Modelling of circuit as flow of data (signal) between ***registers***
- RTL can also be referred to as "data flow" description

- FSM generates control signals
- MUX passes the data based on ***control signals***
- Computation is done on the data path

Functional Specification to RTL



S. Saurabh, "Introduction to VLSI Design Flow". *Cambridge University Press*, 2023.

- Manual Coding: straight forward
- IP Assembly: reusing existing RTL
- Behaviour synthesis: automatic method of generating RTL from high-level language

Pre-RTL Methodologies

Reusing RTL

System-on-chip (SoC) Design Methodology

- Reusing RTL is especially popular in SoC (system-on-chip) design methodologies

System-on-chip (SoC)

- A complete system built on a single chip
- Composed of:
  - Processors, hardware accelerators, memories, peripherals, analog components, and RF devices connected using some structured communication links
  - Embedded software
- Merits:
  - Improves productivity
  - Lowers cost
  - Increases features

Intellectual Properties (IP)

Intellectual Properties (IP)

- Pre-designed and pre-verified sub-systems or blocks
- Can be developed internally or purchased from IP vendors

Content:

- Hardware blocks: processor, memory, interface, etc.
- Software: real-time operating system (RTOS), device drivers, etc.
- Verification IPs (VIPs) eases verification effort

Sharing of Information:

- IPs contain information related to structure, configurability, and interfaces of the subsystem
- Challenge: how to package the information?

Integration of IPs

Integration of IPs (IP Assembly)

- Instantiating various IPs in an SoC and making their connections

Method

- **Metadata:** top-level IP models, bus interfaces, ports, registers, and the required configuration
  - IP-XACT, SystemRDL, XML, or spreadsheet
- **Generator tools:** produce an SoC-level RTL with instantiated IPs.
  - A generator tool can also produce a verification environment and low-level software drivers.

Configuring IPs

- IPs can have configuration parameters such as bus width, power modes, and communication protocols
- IP assembly involves choosing the set of configuration parameters
- **Challenges:** optimality and consistency

Communication Links

- Ad-hoc bus-based
- Structured network on chip (NoC)

Verification Challenges

- Huge functional space
- Software and Hardware

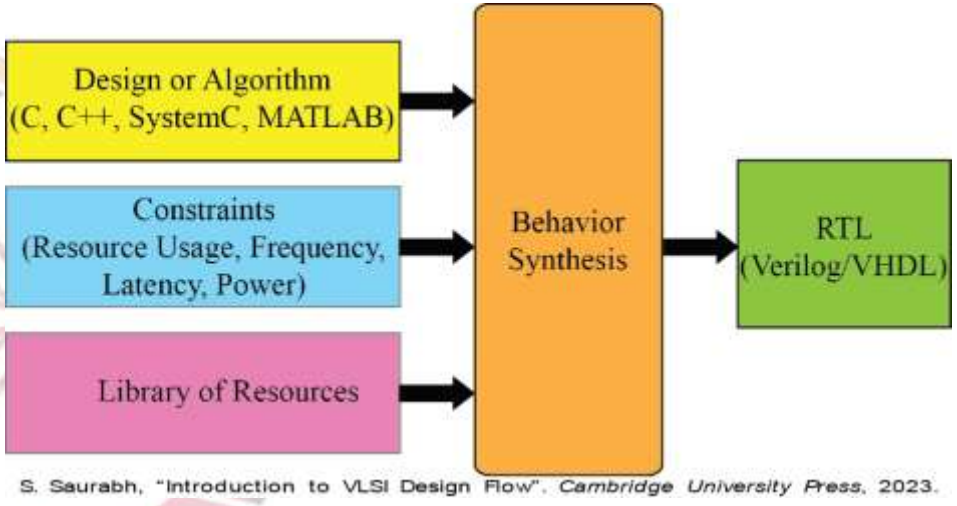
Pre-RTL Methodologies

Behaviour Synthesis

Behavioral Synthesis: What?

Behavioral Synthesis

- Process of converting an algorithm (not timed) to an equivalent design in RTL (fully timed) and satisfy the specified constraints.



S. Saurabh, "Introduction to VLSI Design Flow", *Cambridge University Press*, 2023.

- Behavioral Synthesis is also called High-level Synthesis

Behavioral Synthesis : Cost Metrics (1)

- An untimed algorithm can be implemented in many different ways
- Different implementations can have different cost metrics

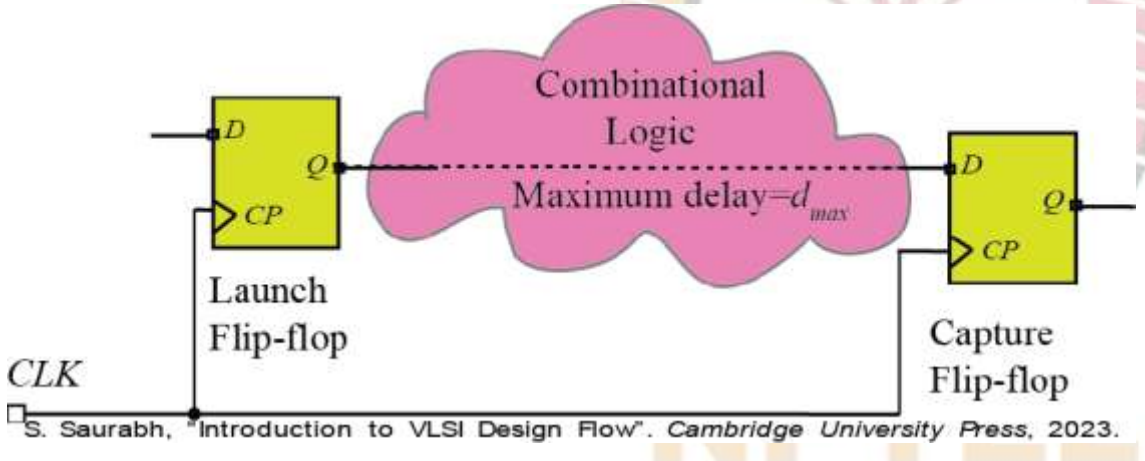
Cost Metrics:

- **Area:** number of circuit elements
- **Latency**: number of clock cycles required before results are available
- **Maximum clock frequency:** worst case combinational delay
- **Power dissipation, Throughput, etc.**

Behavioral Synthesis : Maximum Clock Frequency

Consider a synchronous circuit:

- **Path:** sequence of pins through which a signal can propagate
- **Combinational path:** a path that does not contain any sequential circuit element such as a flip-flop
- **Sequentially adjacent flip-flops:** if the output of one flip-flop is fed as an input to the other flip-flop through a combinational path



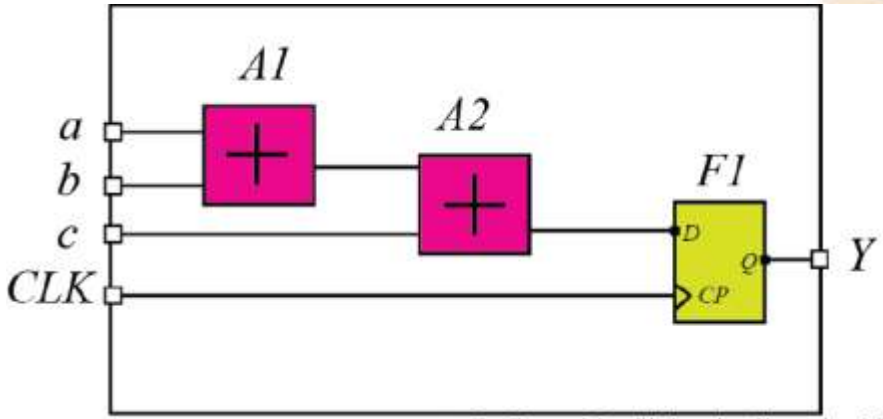
S. Saurabh, "Introduction to VLSI Design Flow". *Cambridge University Press*, 2023.

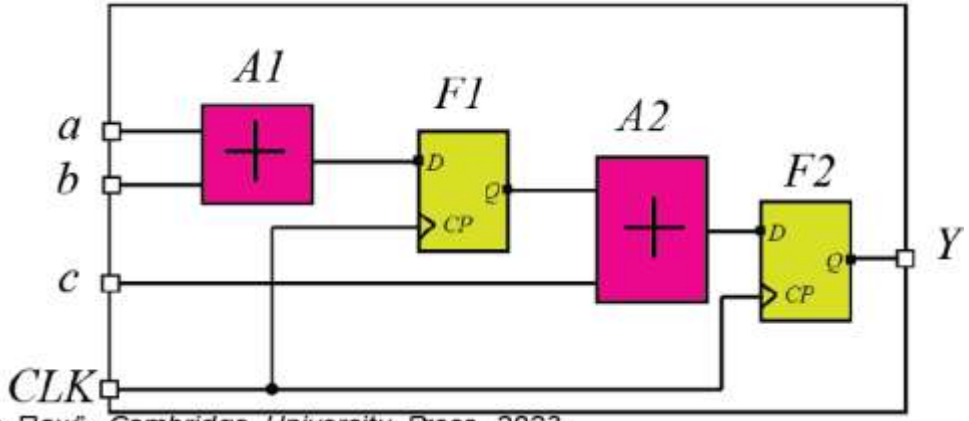
Synchronous circuit: data launched must be captured by the sequentially adjacent flip-flop in the next clock cycle.

- **Clock period** should be greater than the delay of the critical path ($T_p > d_{max}$)
- Maximum clock frequency $f_{max} < 1/d_{max}$
- **Critical Path:** the combinational path that has the largest delay in the circuit (approximately)

Behavioral Synthesis: Illustration (1)

- Algorithmic behavior: $Y = a + b + c$
- Cost metrics: circuit elements used, latency, and maximum delay of combination path.





S. Saurabh, "Introduction to VLSI Design Flow", *Cambridge University Press*, 2023.

Resources
- 2 Adders (+) and 1 Register

Latency
- 1 clock cycle

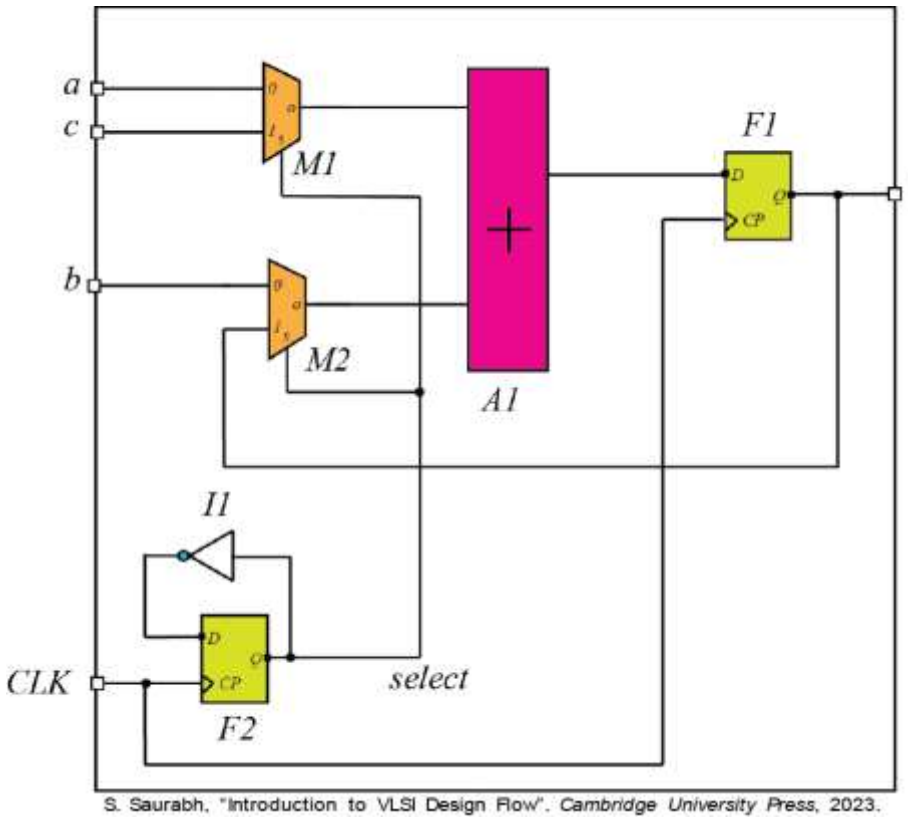
Worst Delay
- Delay of 2 Adders

Resources
- 2 Adders (+) and 2 Registers

Latency
- 2 clock cycle

Worst Delay
- Delay of 1 Adders

Behavioral Synthesis: Illustration (2)



S. Saurabh, "Introduction to VLSI Design Flow", *Cambridge University Press*, 2023.

- Adder is used in the first cycle to compute ($Y = a + b$) and is used in the next cycle to compute ($Y = c + Y$)

- Inputs to adders are controlled by multiplexers
- Multiplexers get "*select*" signal from the control circuitry

Resources
- 1 Adders (+), 2 Register, 2 Multiplexer and 1 Inverter

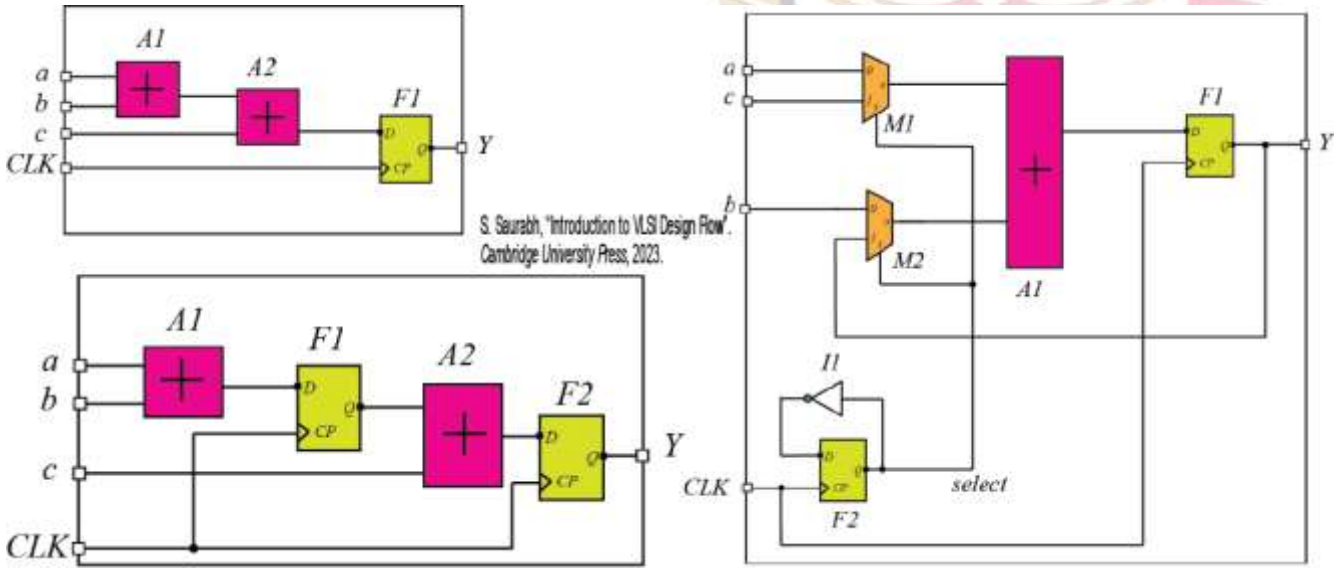
Latency
- 2 clock cycle

Worst Delay
- Delay of 1 Adder + Multiplexer

Behavioral Synthesis: Evaluating Trade-offs

	Area (μm^2)	Delay (ns)
Inverter	1	1
Multiplexer	6	10
Adder	200	100
Flip-flop	12	0

Let us compute the area, latency and critical path delay for three implementations.



	RTL-1	RTL-2	RTL-3
Area	412	424	237
Latency	1 cycle	2 cycle	2 cycle
Delay	200	100	110

Behavioral Synthesis: Untimed to Timed Behavior

Algorithmic behavior: $Y = a + b + c$

Timed Behavior

- Three different "timed" implementation illustrated
- There can be several other implementations
- Behavior synthesis tool will choose the best possible implementation satisfying the constraints

Trade offs

- Behavior synthesis tool can trade off one FoM to improve other FoM

	RTL-1	RTL-2	RTL-3
Area	412	424	237
Latency	1 cycle	2 cycle	2 cycle
Delay	200	100	110

Which RTL will be generated when:

- Area is to be minimized?
- Latency is to be minimized?
- Clock Frequency is to be maximized?

Behavioral Synthesis: Merits and Challenges

Merits:

- Automatic exploration of different possible implementations
  - More exhaustive than handwritten RTL
- Reduces design effort
- Less chance of introducing errors compared to handwritten RTL

Challenges:

- Physical design:
  - Down the flow the QoR may degrade due to other metrics such as congestion not taken into account
- Incremental changes:
  - Lacks readability and debuggability
- Verification challenges

References

- G. D. Micheli. “Synthesis and Optimization of Digital Circuits”. *McGraw-Hill Higher Education*, 1994.
- S. Saurabh, “Introduction to VLSI Design Flow”. Cambridge: *Cambridge University Press*, 2023.