Improved Clock-Gating Control Scheme for Transparent Pipeline

J.H. Choi[1], B.G. Kim[2], A. Dasgupta[3], and K. Roy[2]

[1]SoC Architecture Lab, DMC R&D Center, Samsung Electronics, Korea
[2]Electrical and Computer Eng., Purdue University, W. Lafayette, IN, USA
[3]SoC Enabling Group, Intel Corporation, Austin, TX, USA

Outline

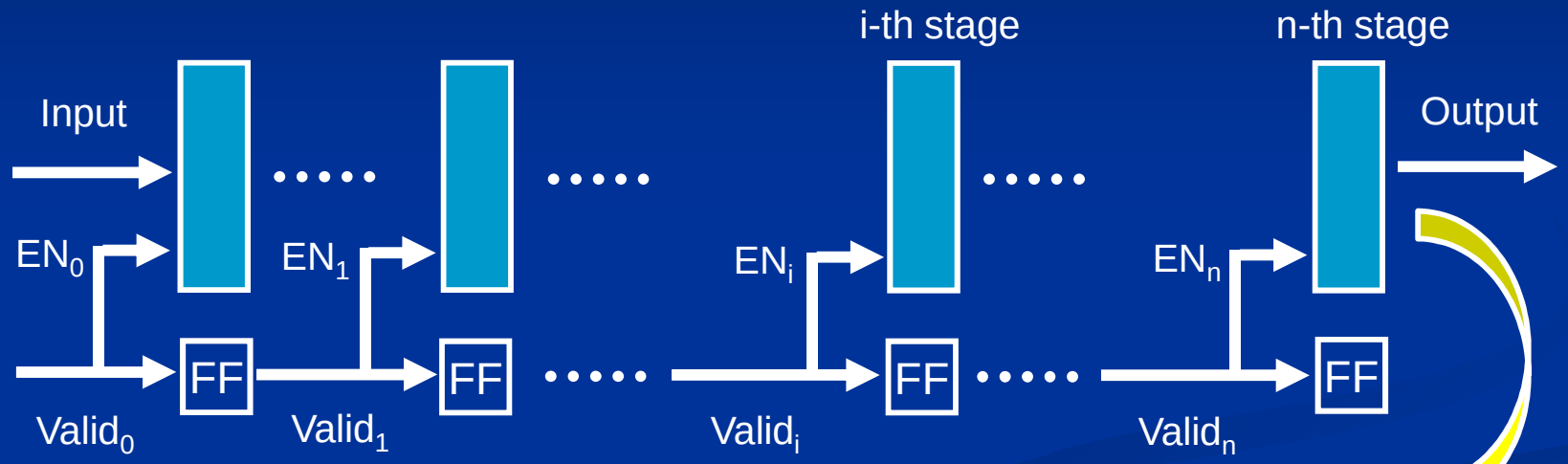
- Introduction
- Previous Works
- Proposed Approach
- Simulation Results
- Conclusion

Introduction

- What is the clock-gating?
  - Reduce power by blocking clock pulses to inactive logic blocks
  - Used for dynamic power reduction
  - Applicable in different levels of design hierarchy from the individual register level to the system level

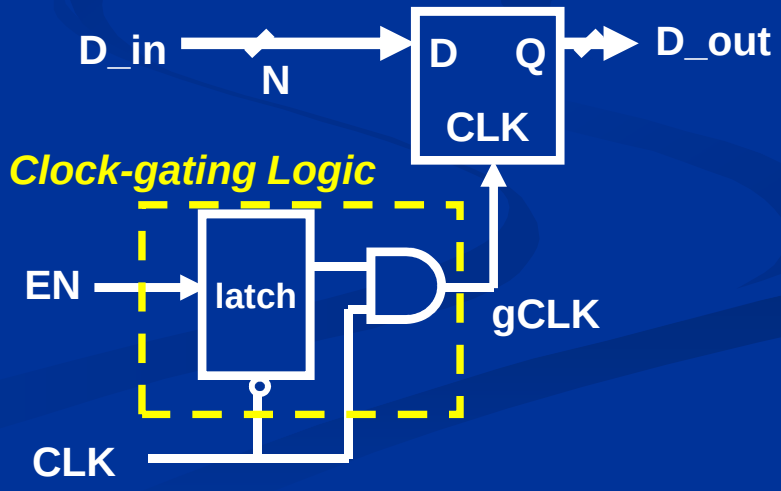
Stage-Level Clock-Gating

- Traditional stage-level clock-gating



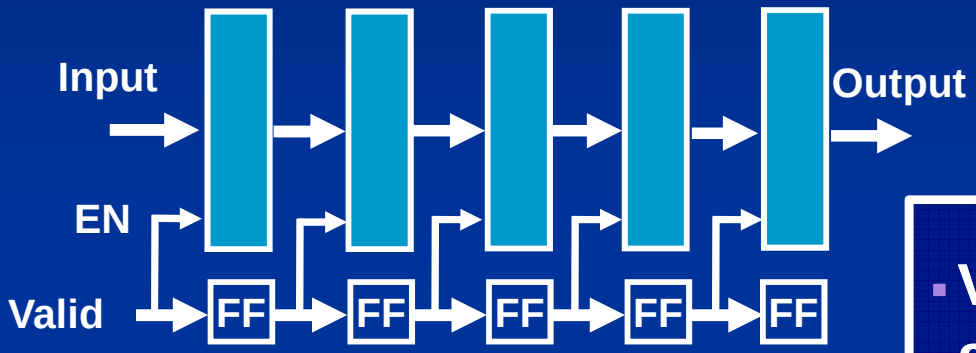
- Clock-gating register
  - EN : clock enable signal
  - Clocked whenever Valid=1

EN	Operation
0	Hold (clock-gated)
1	Normal operation



Stage-Level Clock-Gating (cont.)

- Traditional stage-level clock-gating



Time

(1)	Valid	1		0		0		0	
	Clock		C		U		U		U
(2)	Valid	0		1		0		0	
	Clock		U		C		U		U
(3)	Valid	0		0		1		0	
	Clock		U		U		C		U
(4)	Valid	0		0		0		1	
	Clock		U		U		U		C
(5)	Valid	0		0		0		0	
	Clock		U		U		U		U

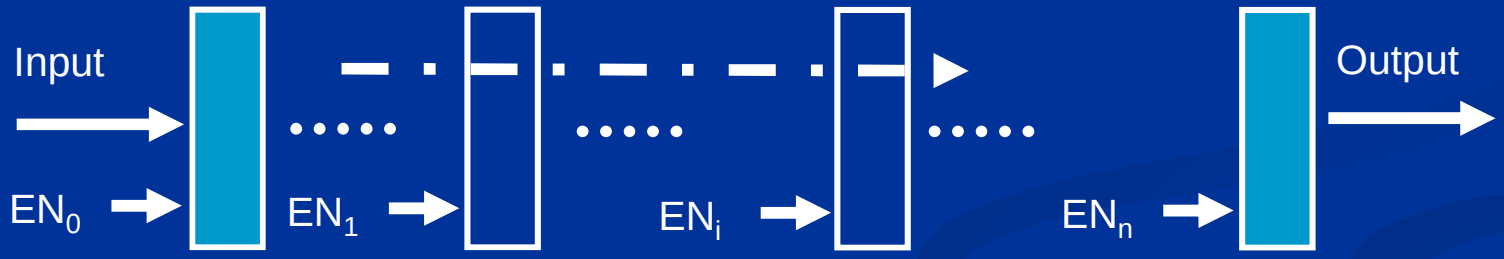
- Valid=1 when new data arrives at the first stage.
- Each stage is clocked when its Valid(=EN) is 1.
- Each stage is clocked one time for one data set.

[C: clocked, U: clock-gated (hold)]

Transparent Pipeline

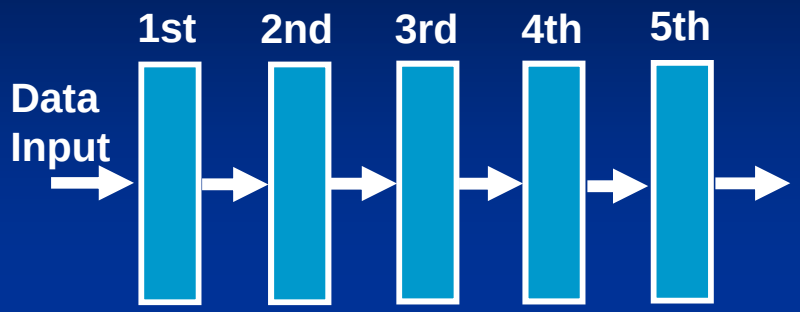
[Jacobson, ISLPED'04]

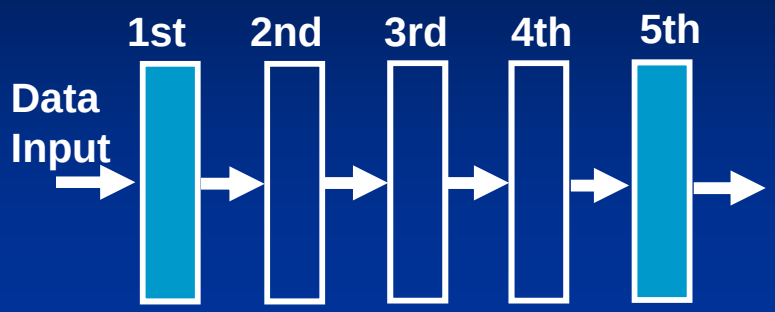
- Transparent pipeline
  - Save clock power by dynamically making registers transparent
  - Intermediate stages are replaced with transparent stages which can selectively bypass data.



  - Example: the 2nd stage bypass the data A driven by the 1st stage.

Transparent Pipeline (cont.)





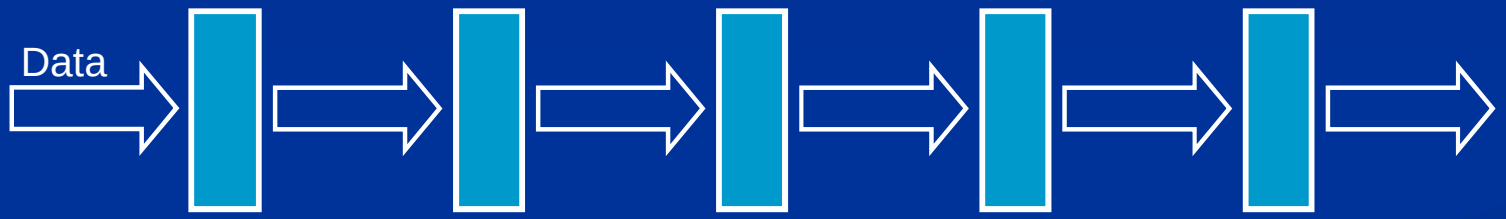
Time Step										
(1)	Valid	1	0	0	0	0	Valid	1	0	0	0	0
	Clock	C	U	U	U	U	Clock	C	T	T	T	U
(2)	Valid	0	1	0	0	0	Valid	0	1	0	0	0
	Clock	U	C	U	U	U	Clock	U	T	T	T	U
(3)	Valid	0	0	1	0	0	Valid	0	0	1	0	0
	Clock	U	U	C	U	U	Clock	U	T	T	T	U
(4)	Valid	1	0	0	1	0	Valid	1	0	0	1	0
	Clock	C	U	U	C	U	Clock	C	T	T	C	U
(5)	Valid	0	1	0	0	1	Valid	0	1	0	0	1
	Clock	U	C	U	U	C	Clock	U	T	T	U	C
(6)	Valid	0	0	1	0	0	Valid	0	0	1	0	0
	Clock	U	U	C	U	U	Clock	U	T	T	U	U
(7)	Valid	0	0	0	1	0	Valid	0	0	0	1	0
	Clock	U	U	U	C	U	Clock	U	T	T	T	U
(8)	Valid	0	0	0	0	1	Valid	0	0	0	0	1
	Clock	U	U	U	U	C	Clock	U	T	T	T	C

[C: clocked, U: clock-gated (hold), T: transparent (bypassing)]

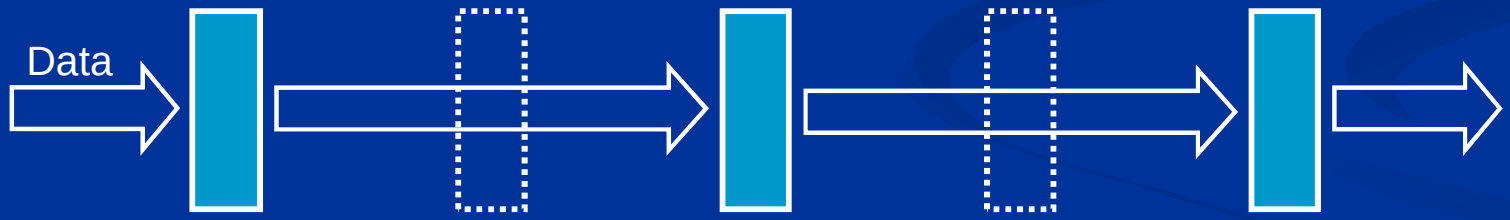
Collapsible Pipeline

[Shimada et al., ISLPED'03]

- Adjust pipeline depth with *transparent* stages
  - Normal operation with freq. = F



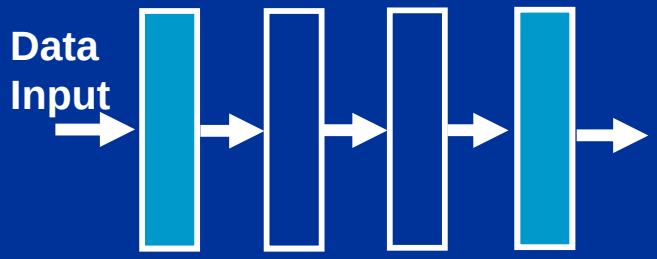
  - Low-power operation (shallow mode) with freq. = F/2

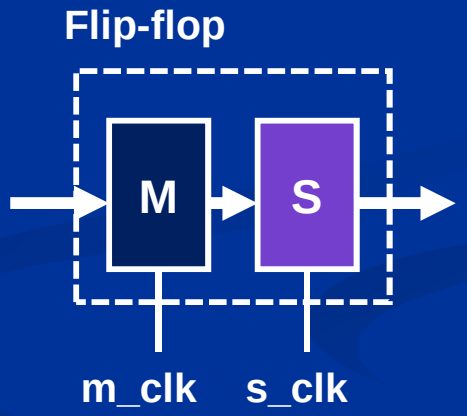


- Clock power saving in *shallow* mode
- No power reduction in normal operation

Contributions

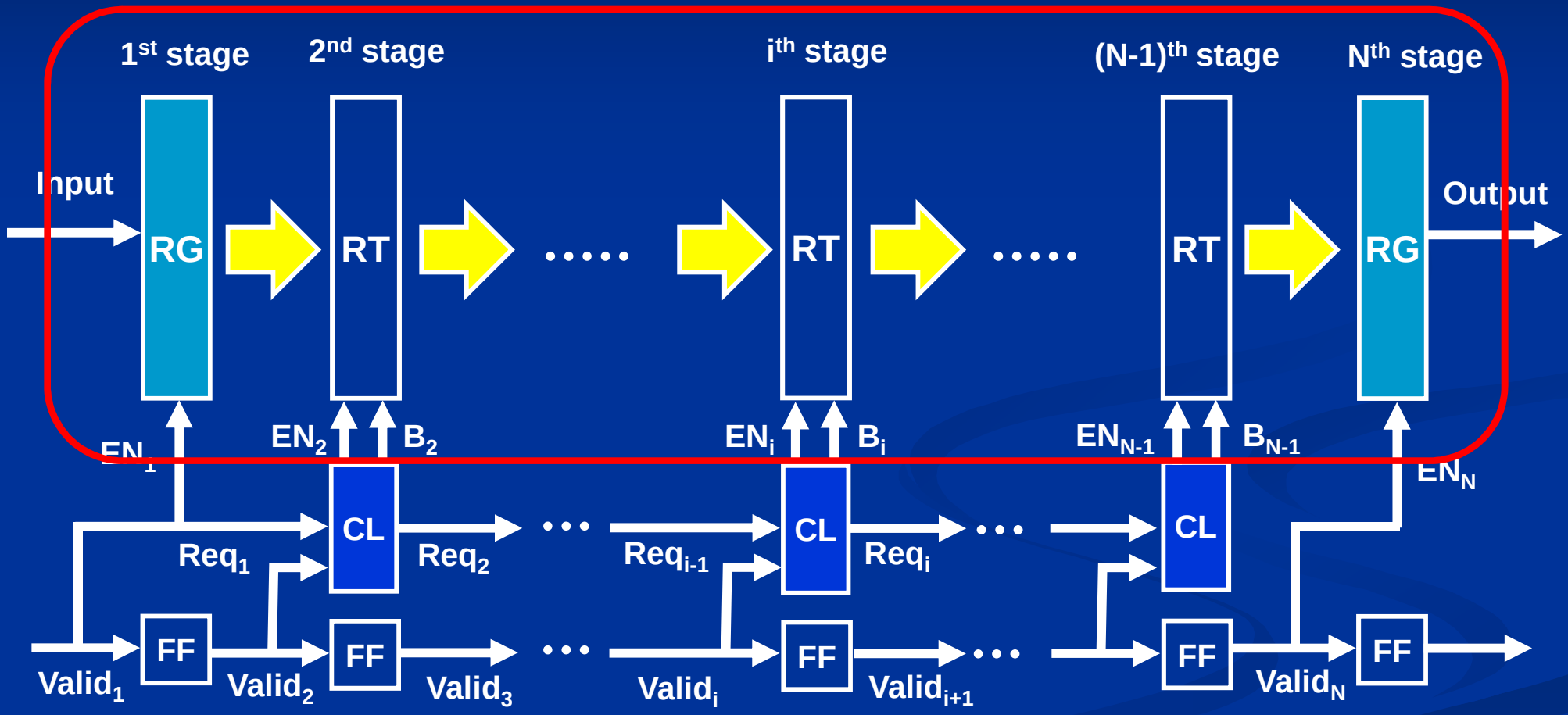
- Previous transparent pipeline [Jacobson, ISLPED'04]
  - Limited to two consecutive transparent stages
  - Transparent stages need two separate clock lines for separate control of master and slave latch.





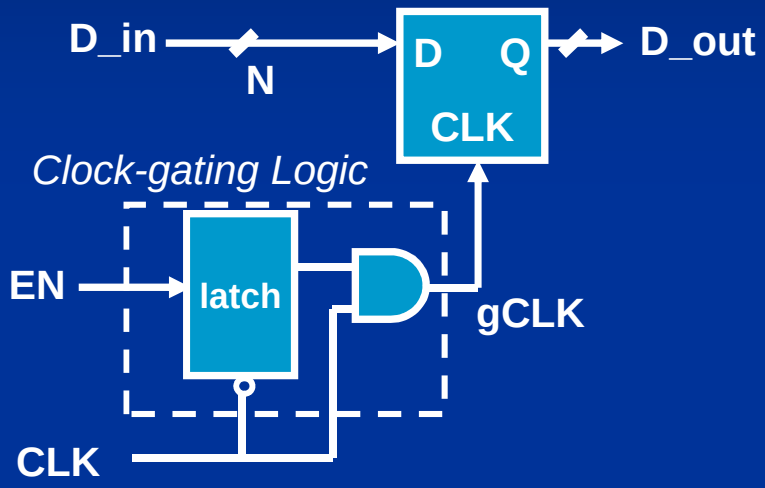
- Proposed approach
  - Improved control logic for transparent pipeline
    - Applicable to any number of pipeline stages
    - Easily extended for pipeline collapsing
  - Low overhead FF for transparent mode

Proposed Transparent Pipeline



Pipeline Registers

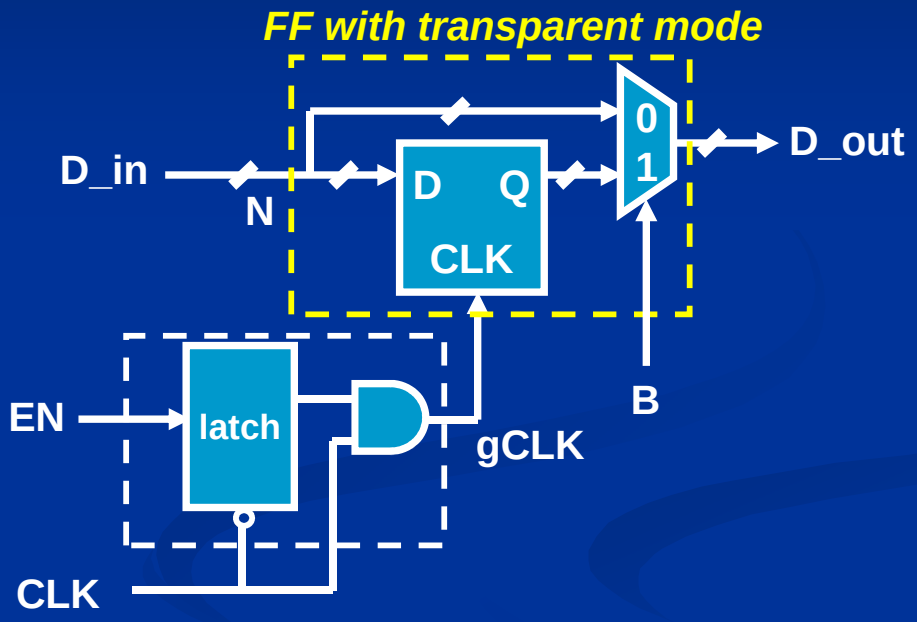
- Normal (opaque) stage



EN	Operation
0	Hold (clock-gated)
1	Normal operation

- EN: Clock enable signal
- B: Transparent state bit (0: transparent, 1: opaque)

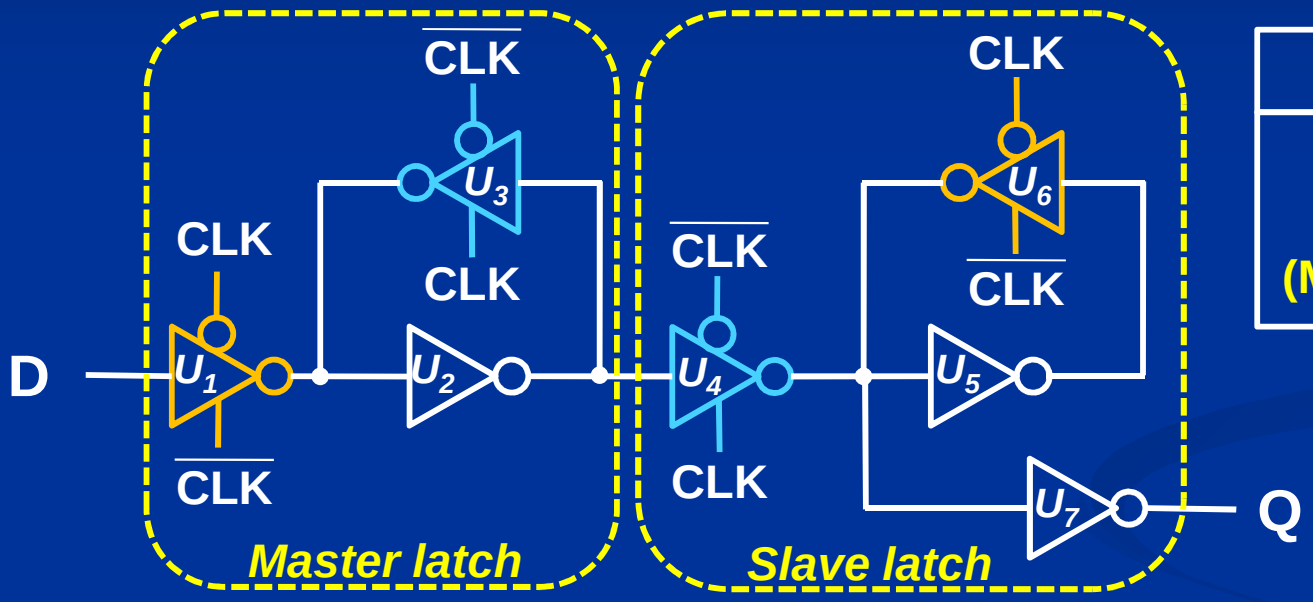
- Transparent stage



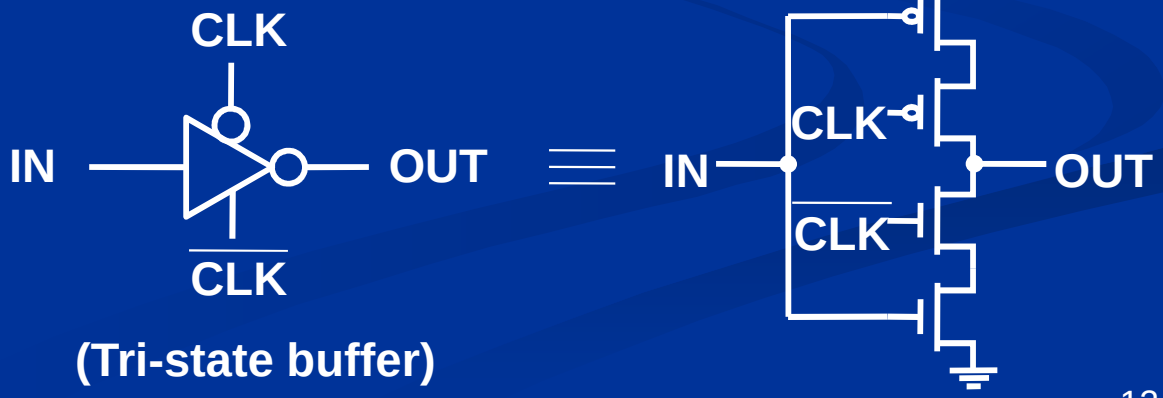
EN	B	Operation
0	0	Bypass (transparent)
0	1	Hold (clock-gated)
1	1	Normal operation

Pipeline Registers (cont.)

- Schematic of static D-FF

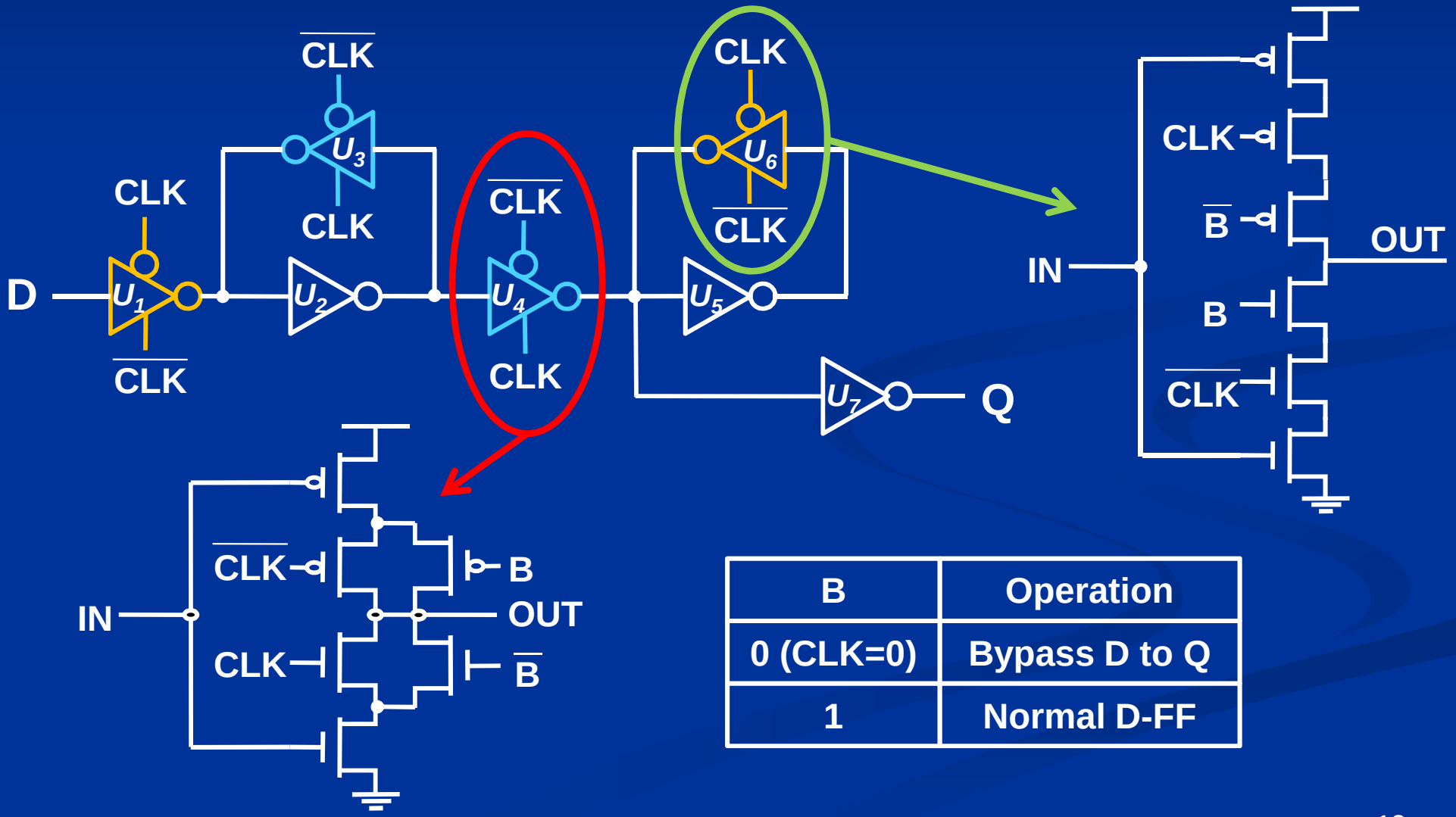


CLK=0	CLK=1
U1/U6 on U3/U4 off (M.Latch on)	U1/U6 off U3/U4 on (S.Latch on)



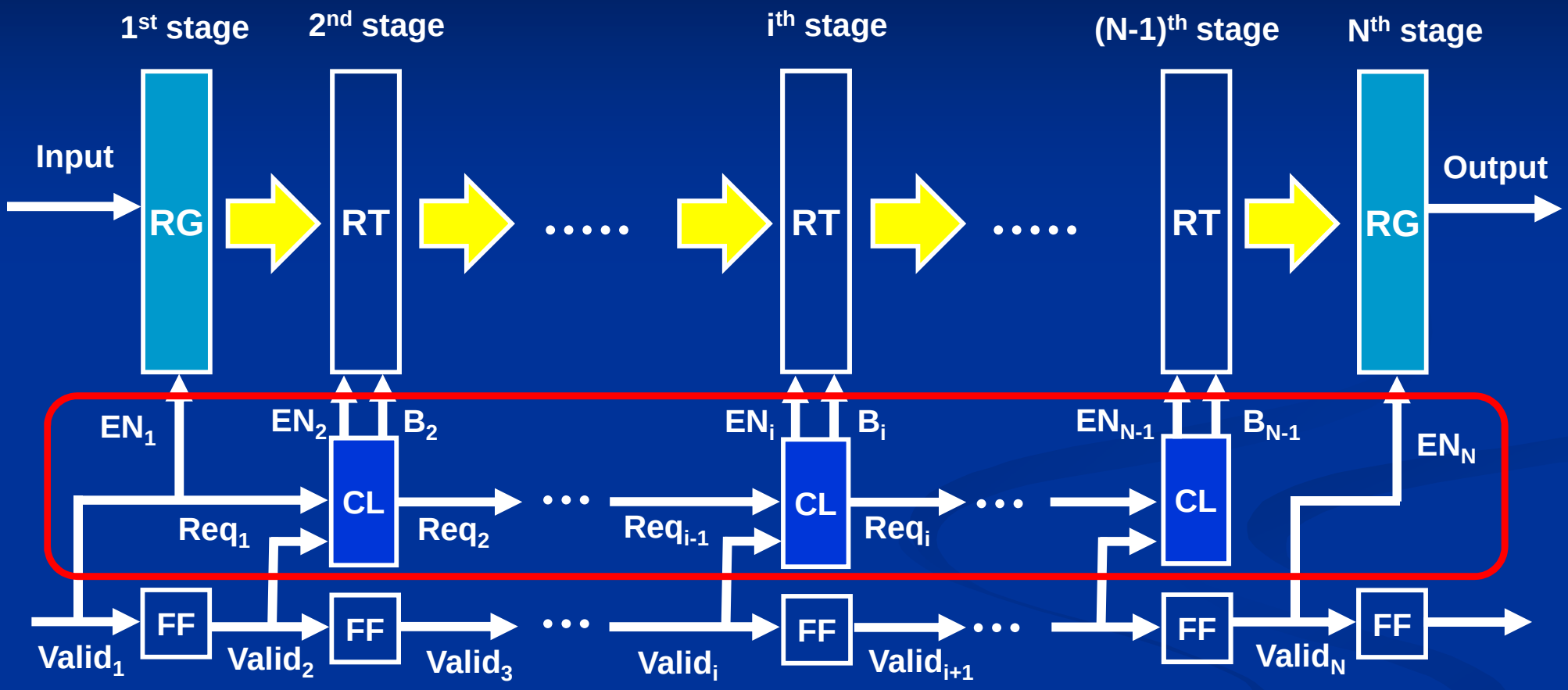
Pipeline Register (cont.)

- D-FF with transparent mode



B	Operation
0 (CLK=0)	Bypass D to Q
1	Normal D-FF

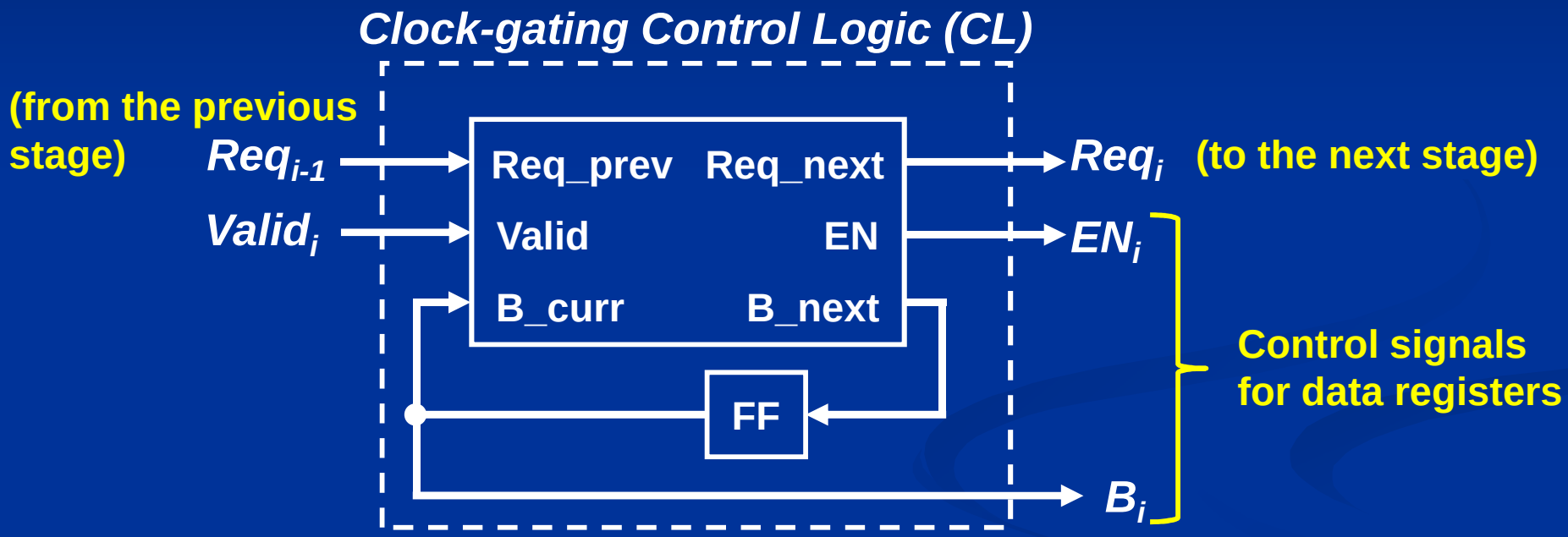
Proposed Transparent Pipeline



- **Criteria for correct functionality** (to avoid race conditions): At least one opaque stage between two different data sets
- **Req signal**: Generated when a stage needs to accept new data set, to notify that other stage in the downstream should take the data.

Control Logic

- Control logic for transparent stage registers

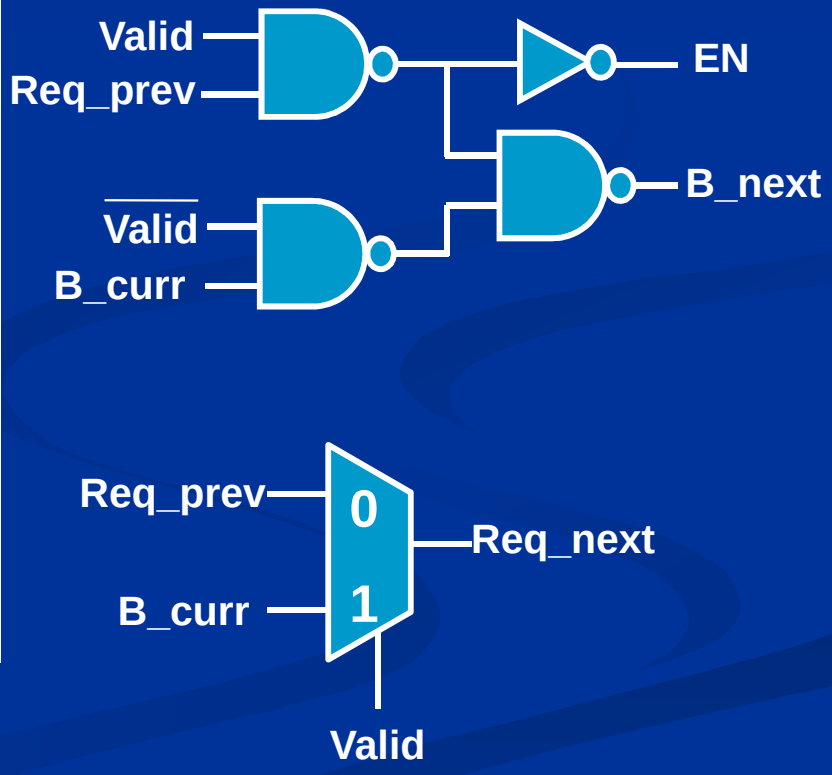


Inputs	Outputs
B_curr: current state (0: transparent)	**B_next**: next state
Valid: data valid bit	**EN**: clock enable signal
Req_prev: from the prev stage	**Req_next**: to the next stage

Control Logic (cont.)

- Truth table and gate-level implementation

Inputs			Outputs		
B_curr	Valid	Req_prev	B_next	EN	Req_next
0	0	0	0	0	0
1	0	0	1	0	0
0	0	1	0	0	1
0	1	1	1	1	0
1	1	1	1	1	1
0	1	0	0	0	0
1	1	0	0	0	1
1	0	1	x (1)	x (0)	x (1)



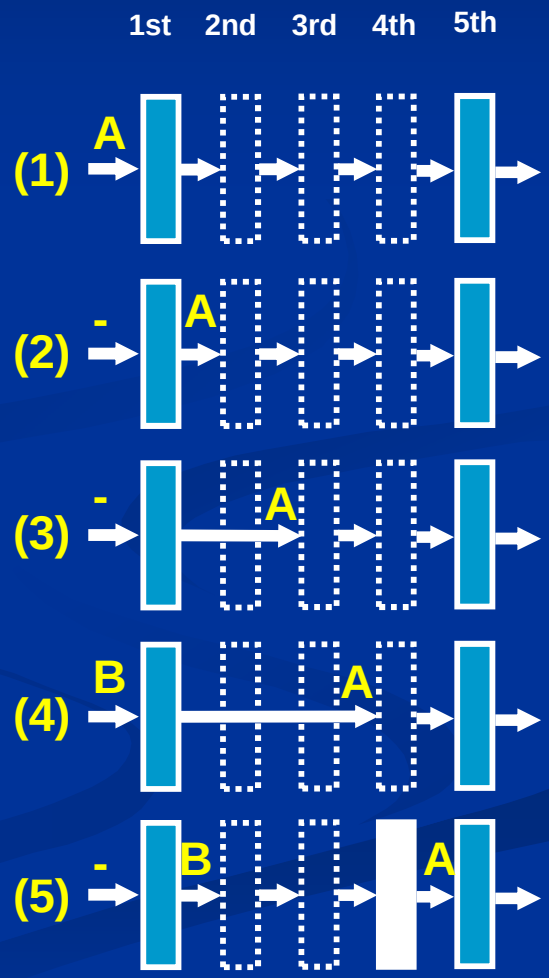
Transparent stage is clocked (EN=1) when both Valid and Req_prev are 1.

Control Logic (cont.)

- Working example in 5-stage pipeline

Time step	1st (opaque)			2nd (trans)					3rd (trans)					4th (trans)					5th (opaque)		
	V	E	R+	V	B-	B+	E	R+	V	B-	B+	E	R+	V	B-	B+	E	R+	V	E	R+
(1)	**1**	**1**	1	**0**	0	0	0	1	**0**	0	0	0	1	**0**	0	0	0	1	**0**	0	0
(2)	**0**	0	0	**1**	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	0	0
(3)	**0**	0	0	**0**	0	0	0	0	**1**	0	0	0	0	**0**	0	0	0	0	**0**	0	0
(4)	**1**	**1**	1	**0**	0	0	0	1	**0**	0	0	0	1	**1**	0	1	**1**	0	**0**	0	0
(5)	**0**	0	0	**1**	0	0	0	0	**0**	0	0	0	0	**0**	1	1	0	0	**1**	**1**	1
(6)	**0**	0	0	**0**	0	0	0	0	**1**	0	0	0	0	**0**	1	1	0	0	**0**	0	0
(7)	**0**	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**1**	1	0	0	1	**0**	0	0
(8)	**0**	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**0**	0	0	0	0	**1**	**1**	1

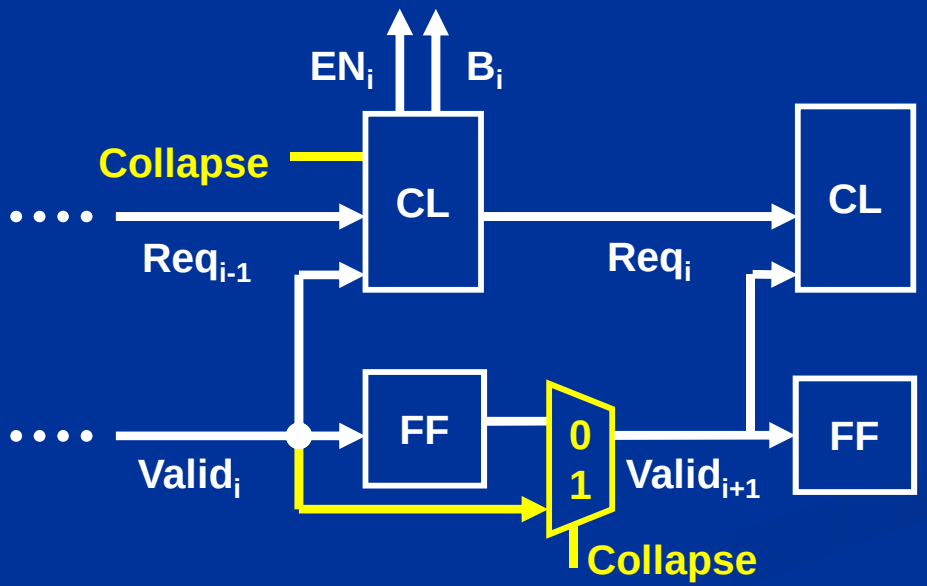
[V: Valid, E: Enable, B-: B_curr, B+: B_next, R+: Req_next]

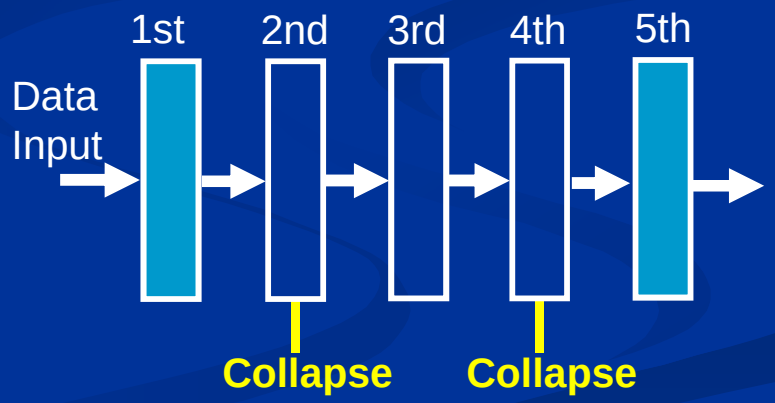


Control Logic (cont.)

- Extension for collapsible pipeline

Inputs				Outputs		
Collapse	B_curr	Valid	Req_prev	B_next	EN	Req_next
0	Same as previous			Same as previous		
1	x	x	x	0	0	Req_prev

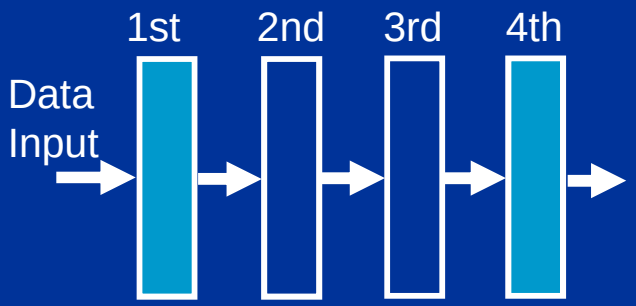


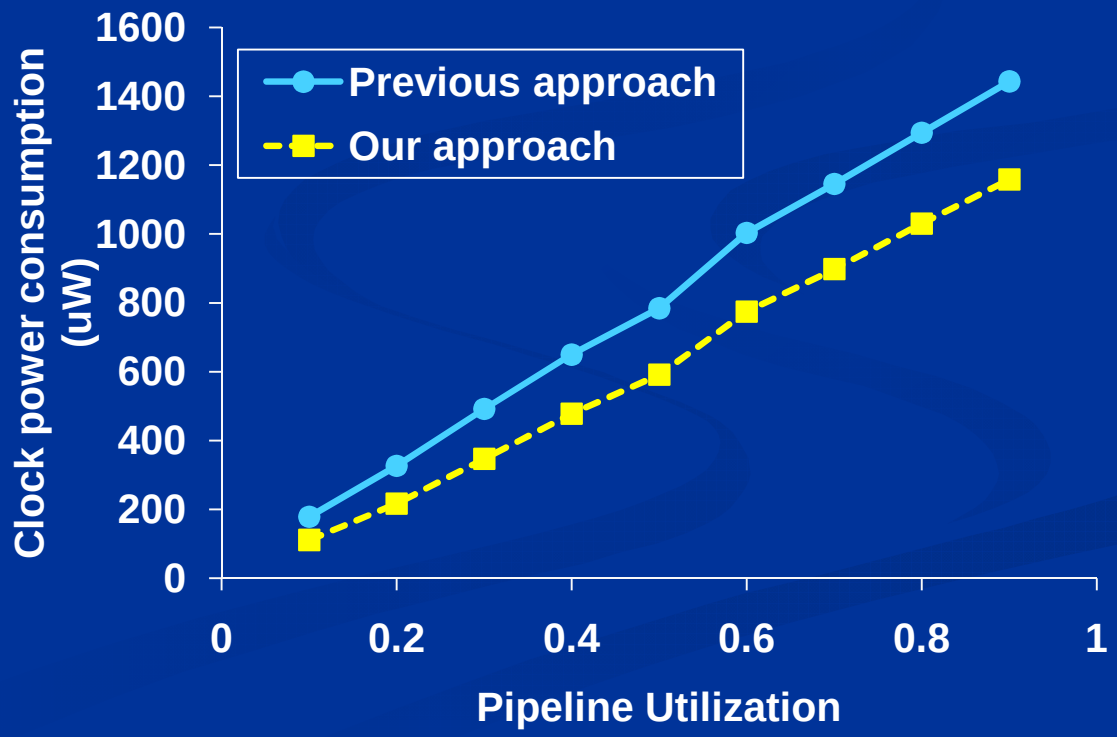


5 stages in normal mode
3 stages in shallow mode

Simulation Results

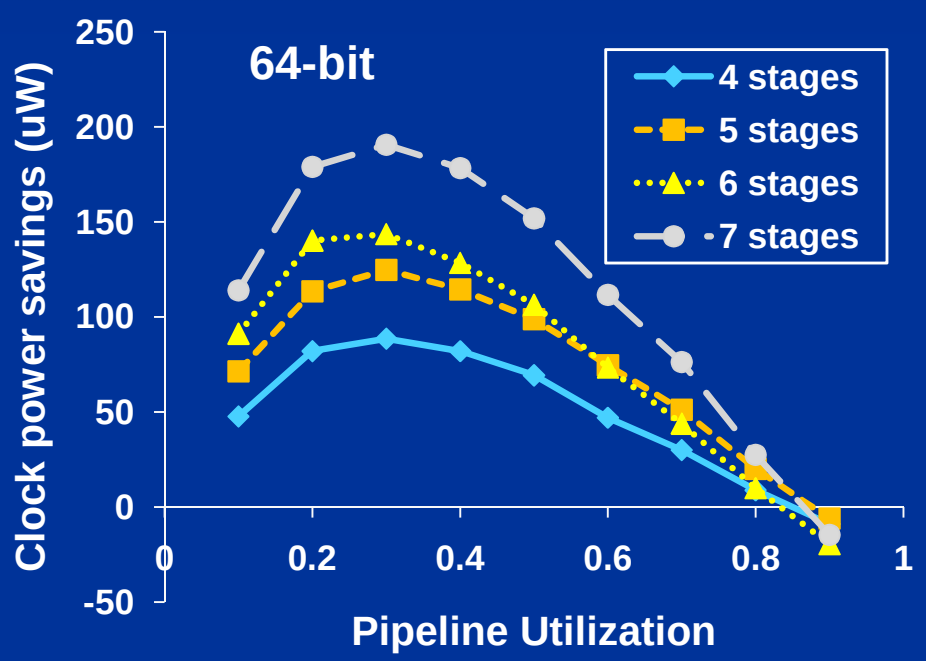
- Comparison with the previous transparent pipeline technique [Jacobson, ISLPED'04]
  - Vdd=1.2V, f=500MHz, IBM 90nm technology
  - Pipeline utilization varies from 0.1 to 0.9.
  - 64-bit 4-stage pipeline

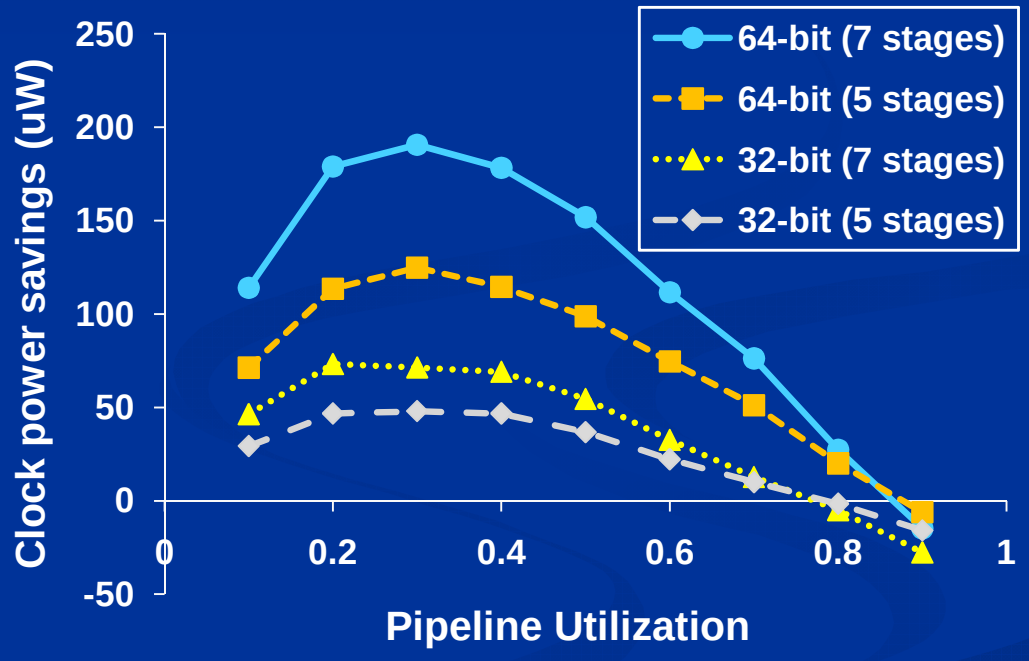




Simulation Results (cont.)

- Power saving over traditional stage-level clock-gating
  - Various pipeline depth and width are considered.
  - Power overhead of control logics is included.

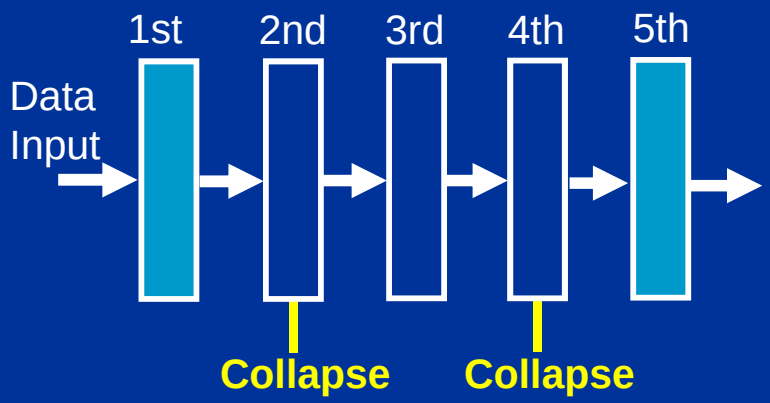




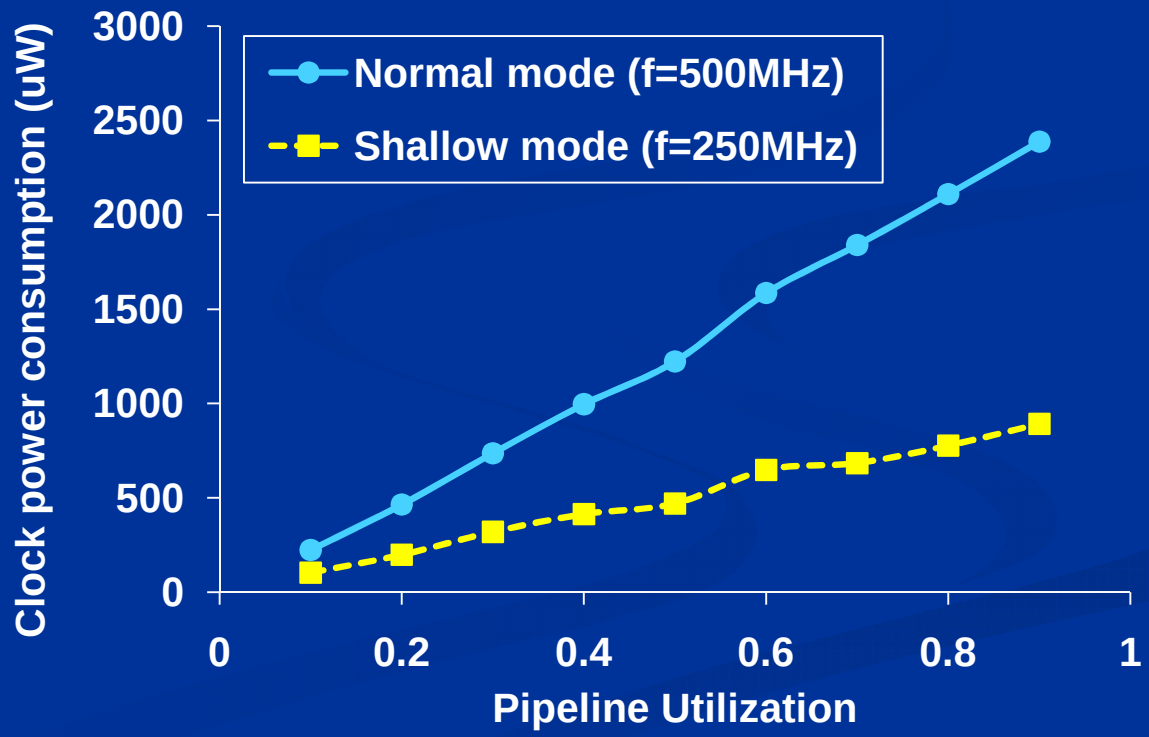
- **More power saving from wider and deeper pipelines**
- **At high utilization (0.9), the proposed approach consumes more power due to control overhead and increased glitch.**

Simulation Results (cont.)

- Extension to collapsible pipeline
  - 64-bit 5-stage pipeline (2nd and 4th stages are collapsible)
  - f_{normal}=500MHz, $f_{shallow}$=250MHz



- **In shallow mode, clock power decreases to ~1/3 of normal mode.**



Summary

- Clock-gating technique for dynamic power reduction
  - Compact control logic for transparent pipeline
  - Applicable to any number of pipeline stages
  - Transparent FF with low hardware overhead
  - Energy/performance trade-off through pipeline stage collapsing