



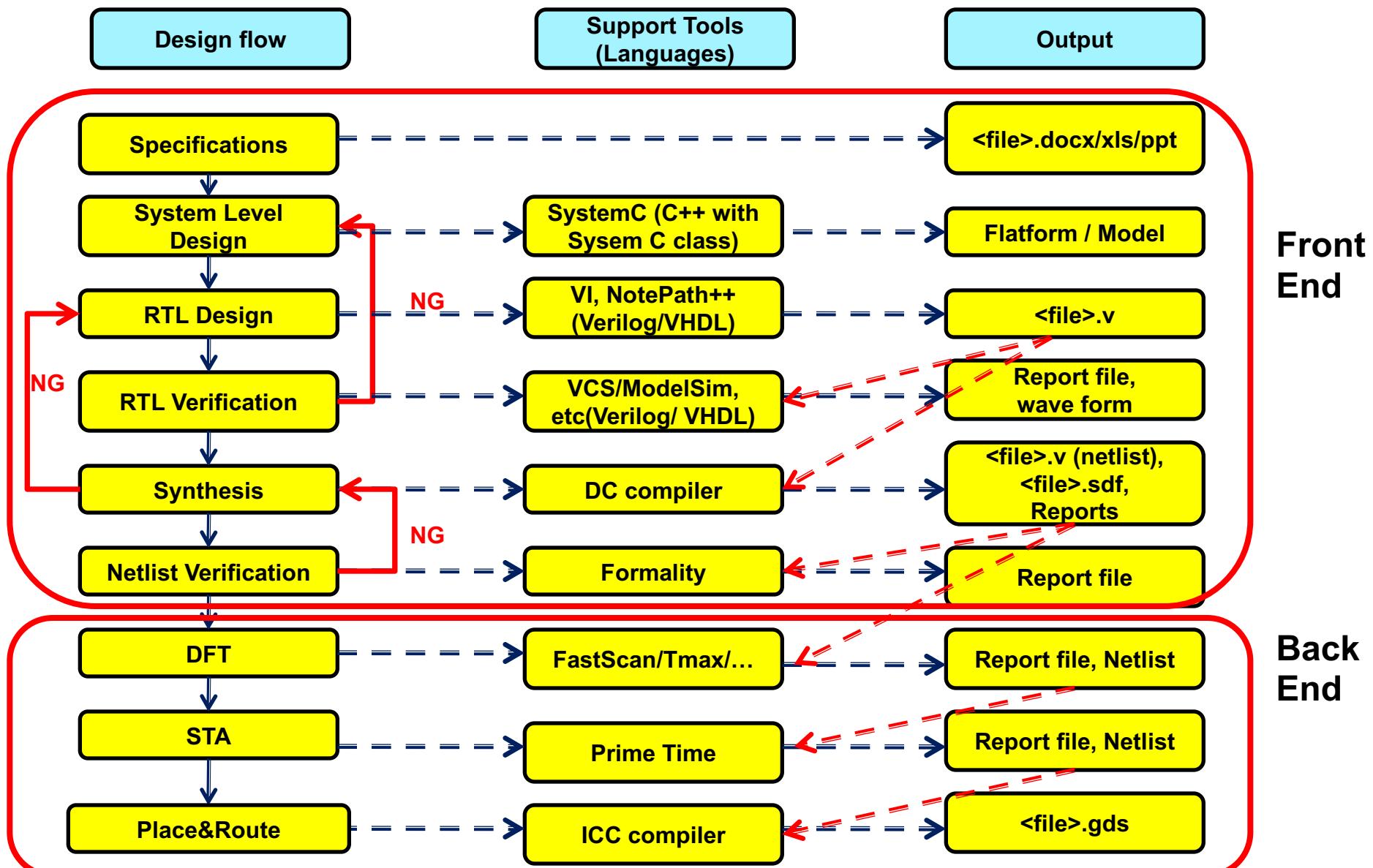
Specification for Cell Based Design Flow



Cell Based Design & Specification

- ❖ Cell Based Design Flow
- ❖ How To Start A Specification
- ❖ Specification Examples
- ❖ Controller and Data Hazard

Cell Based Design Flow



Cell Based Design Flow

Design flow

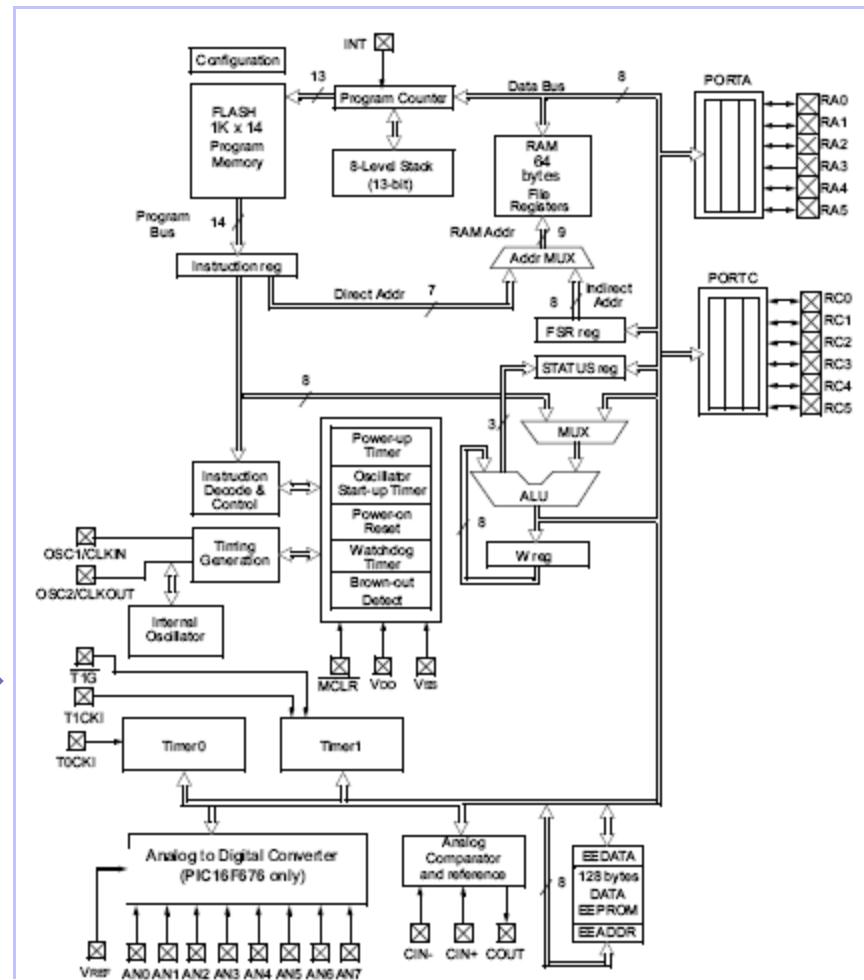
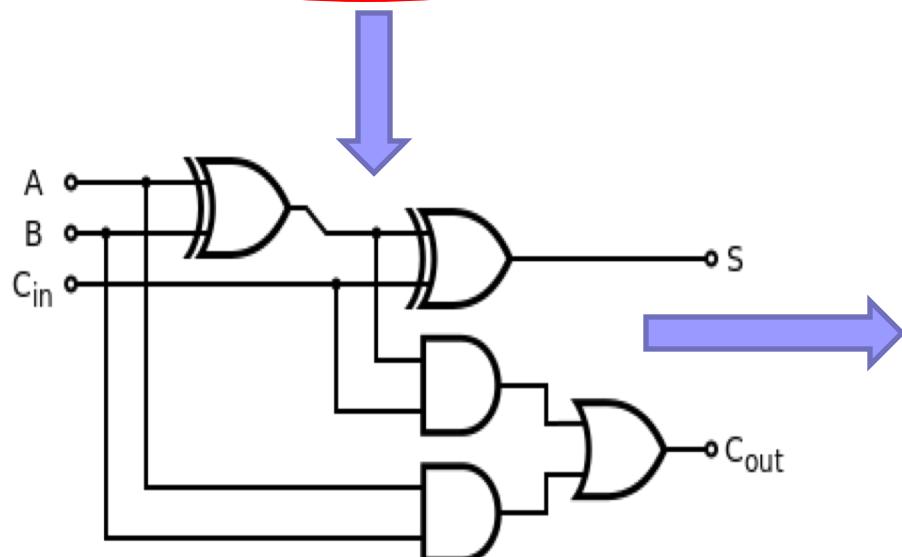
Support Tools
(Languages)

Output

Specifications

<file>.docx/xls/ppt

Design the one bit adder:
+ Three inputs (A, B, Cin)
+ Two output (S, Cout)





Cell Based Design & Specification

- ❖ Cell Based Design Flow
- ❖ How To Start A Specification
- ❖ Specification Examples
- ❖ Controller and Data Hazard

How to start specification ?

Specifications

Data

DSP

Core

IPs

Bus

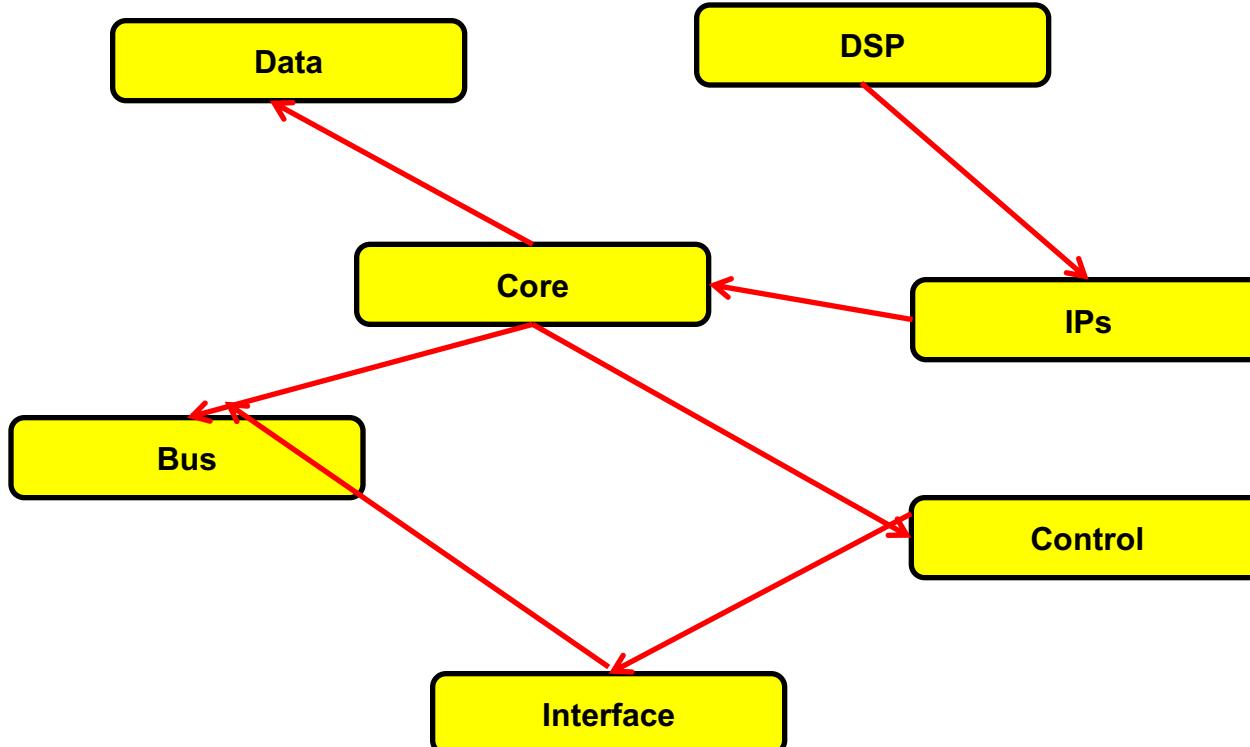
Control

Interface



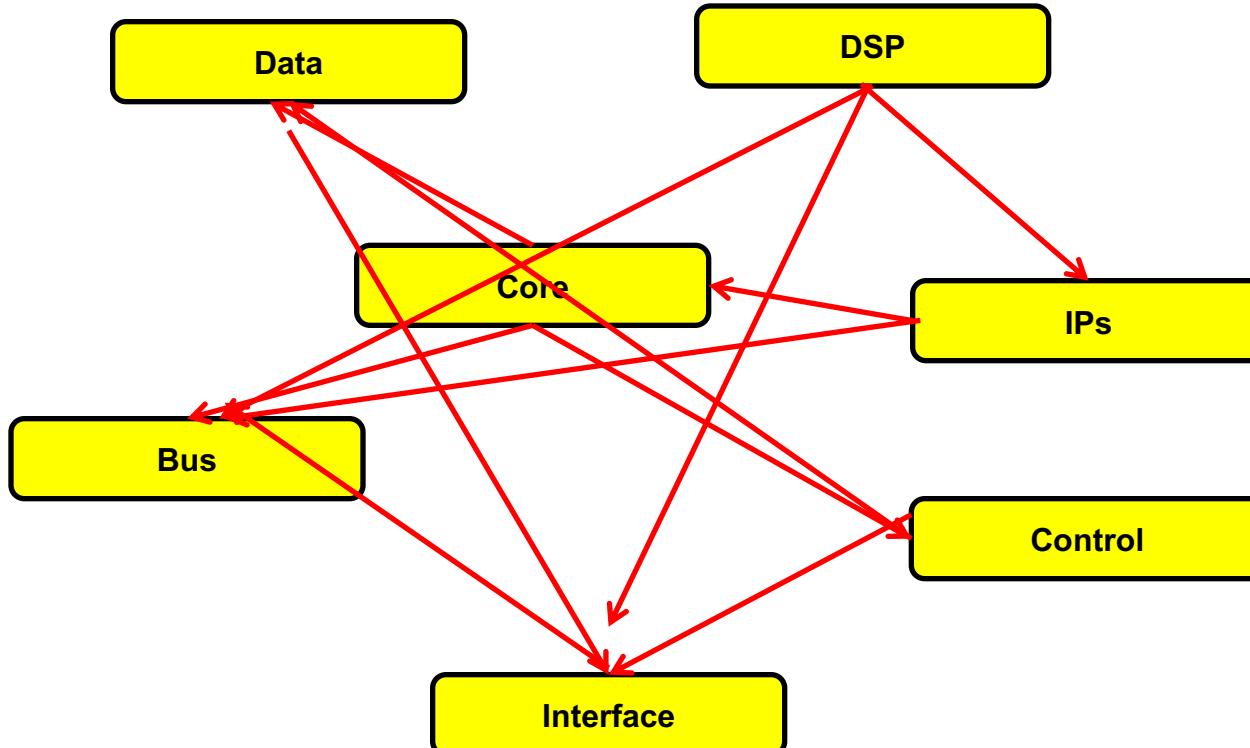
How to start specification ?

Specifications



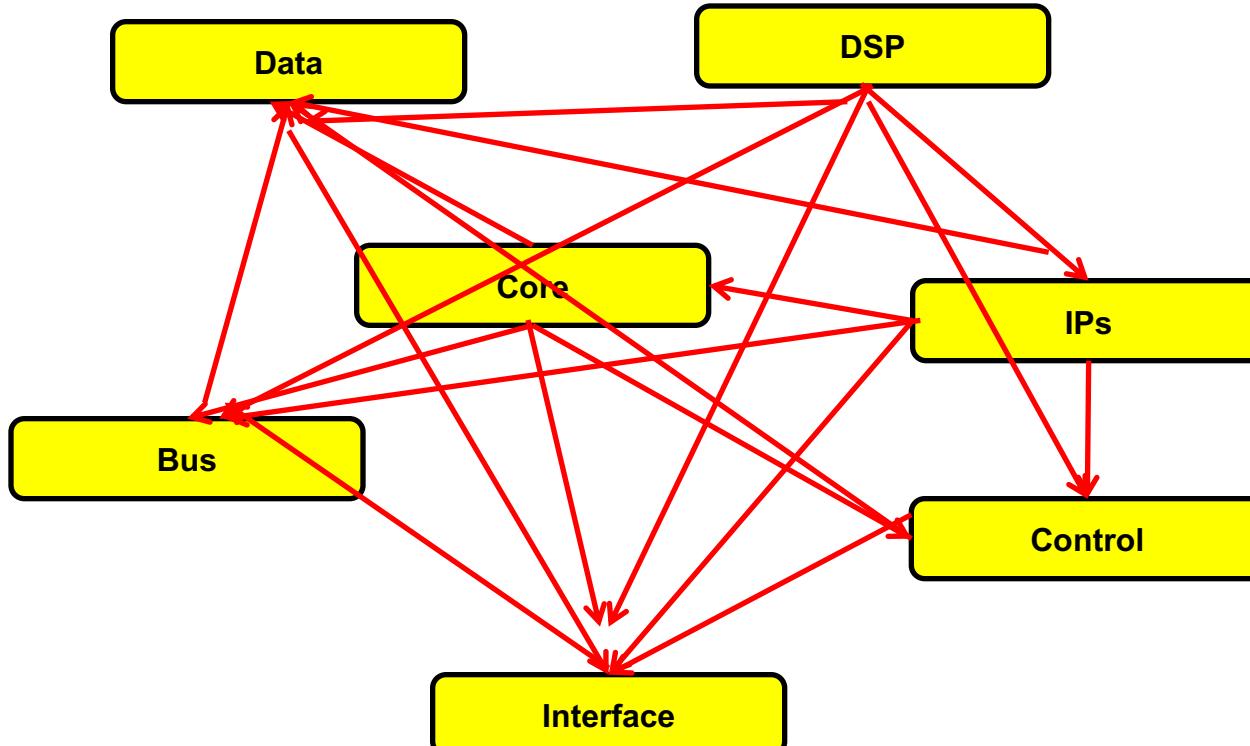
How to start specification ?

Specifications



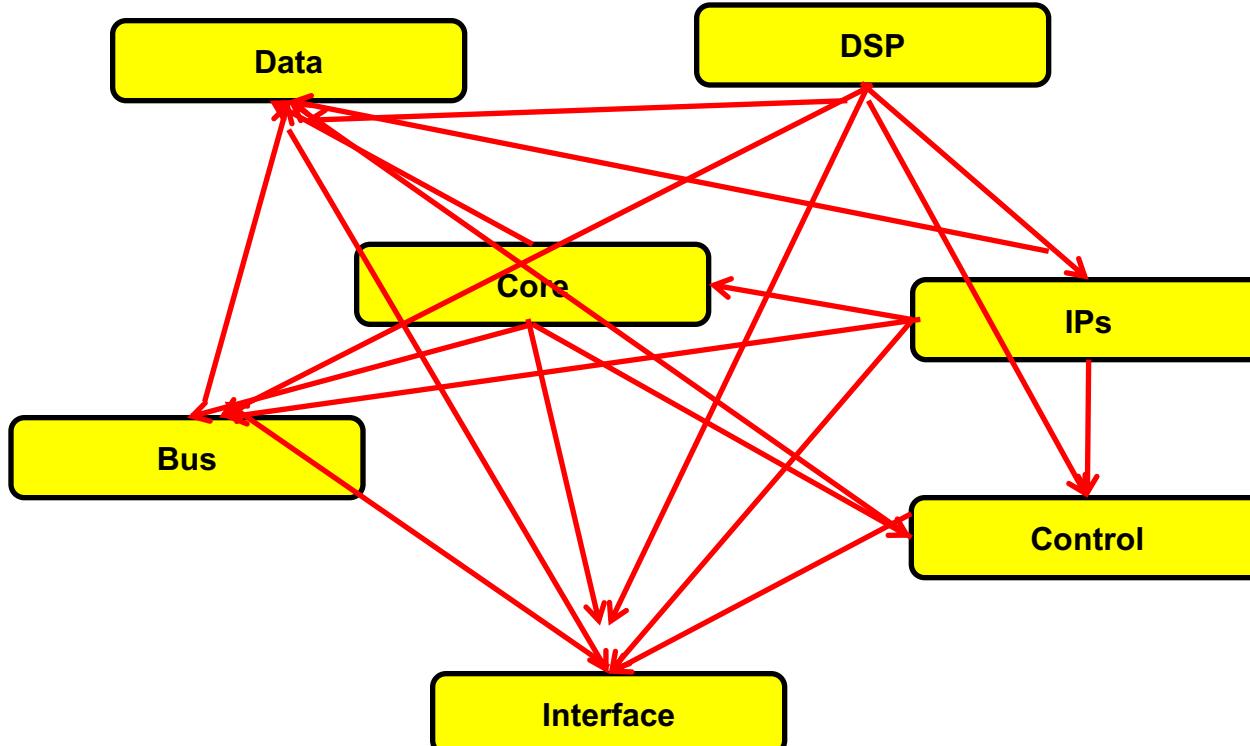
How to start specification ?

Specifications



How to start specification ?

Specifications



???



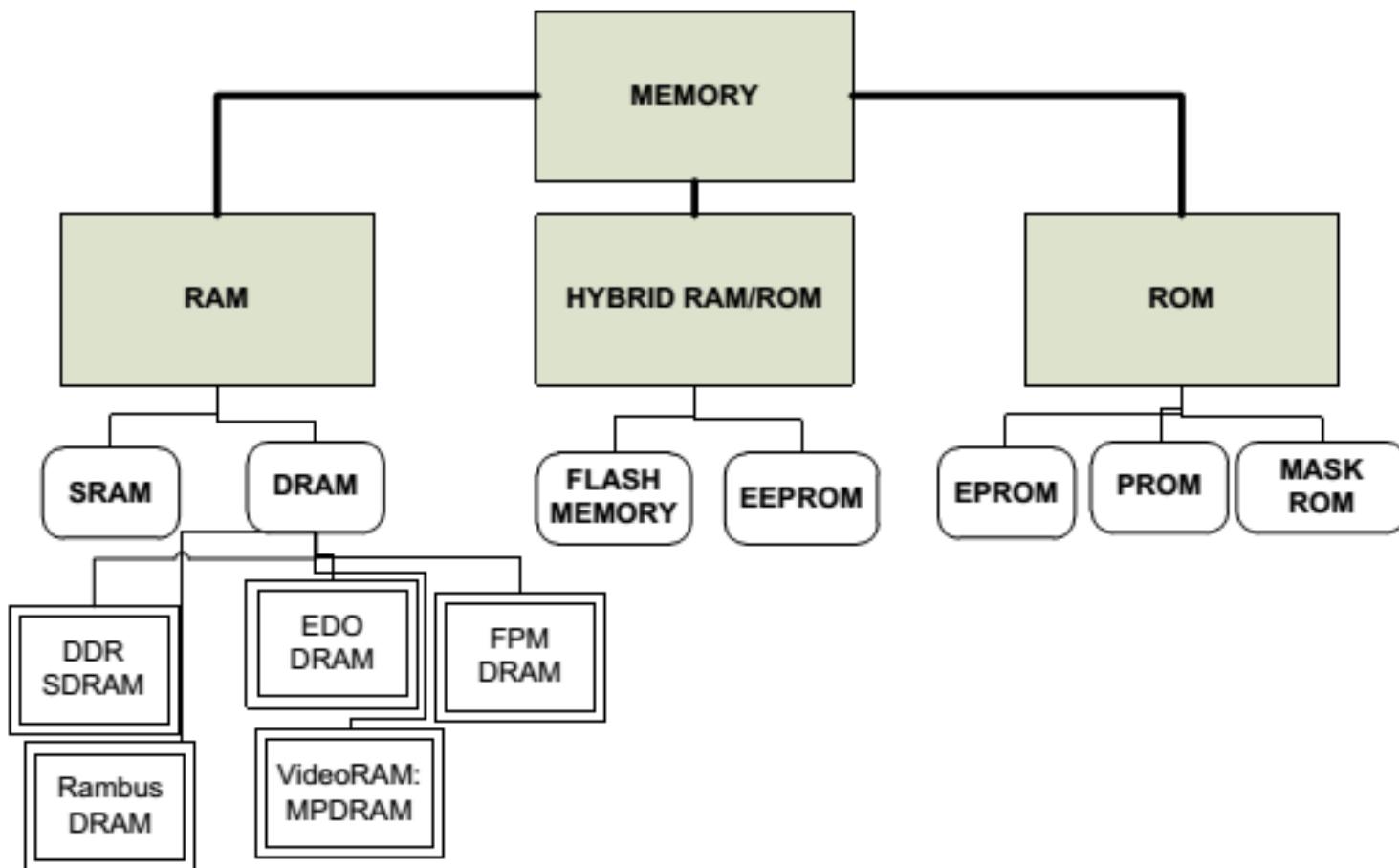
Cell Based Design & Specification

- ❖ Cell Based Design Flow
- ❖ How To Start A Specification
- ❖ Specification Examples
- ❖ Controller and Data Hazard

Memory Specification

Specifications

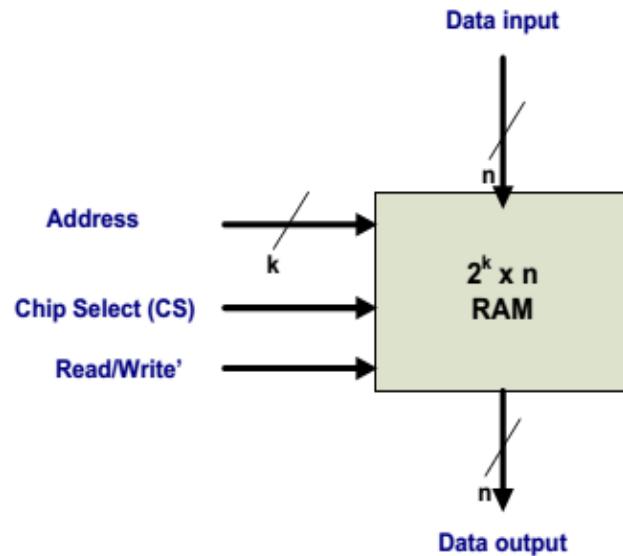
Memory Types



Memory Specification

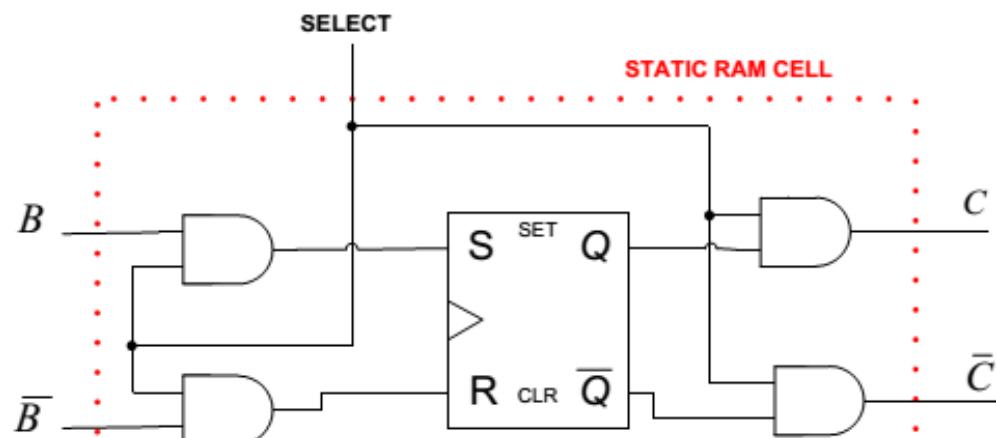
Specifications

This block diagram introduces the main interface to RAM.



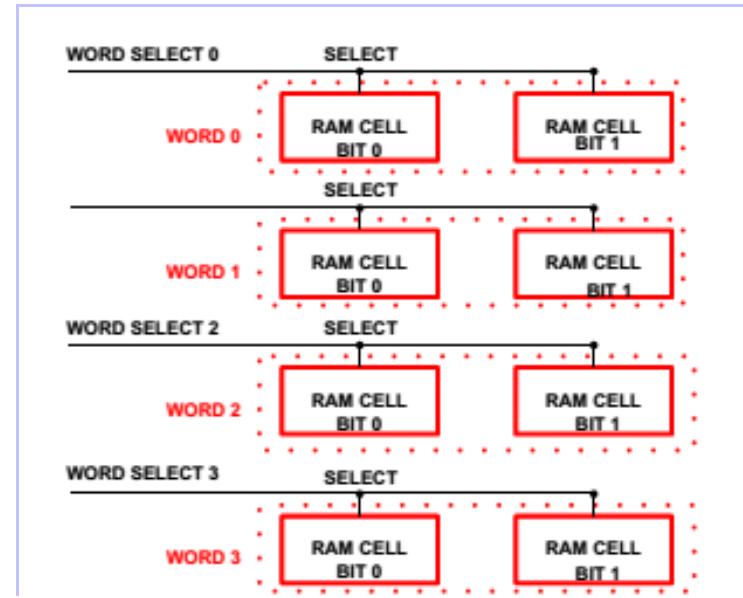
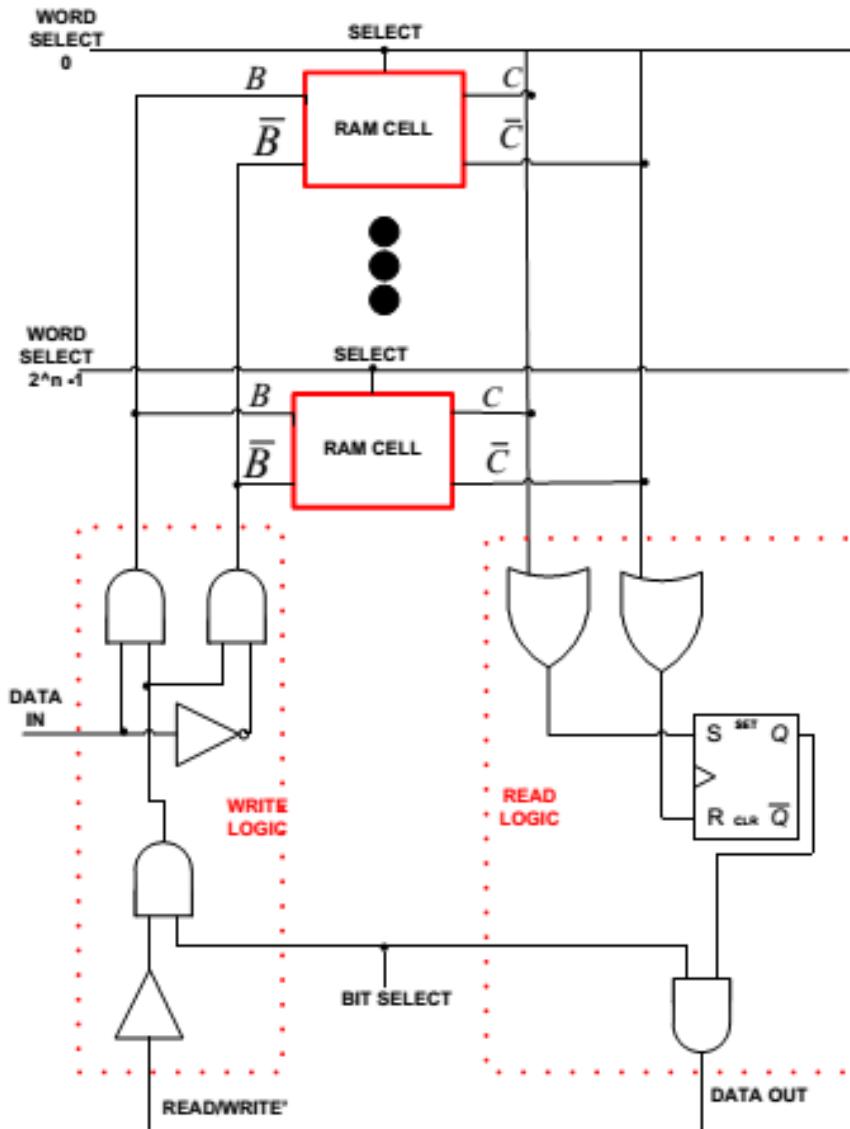
CS	R/W'	Operation
0	X	No operation
1	0	write
1	1	read

Static RAM cell



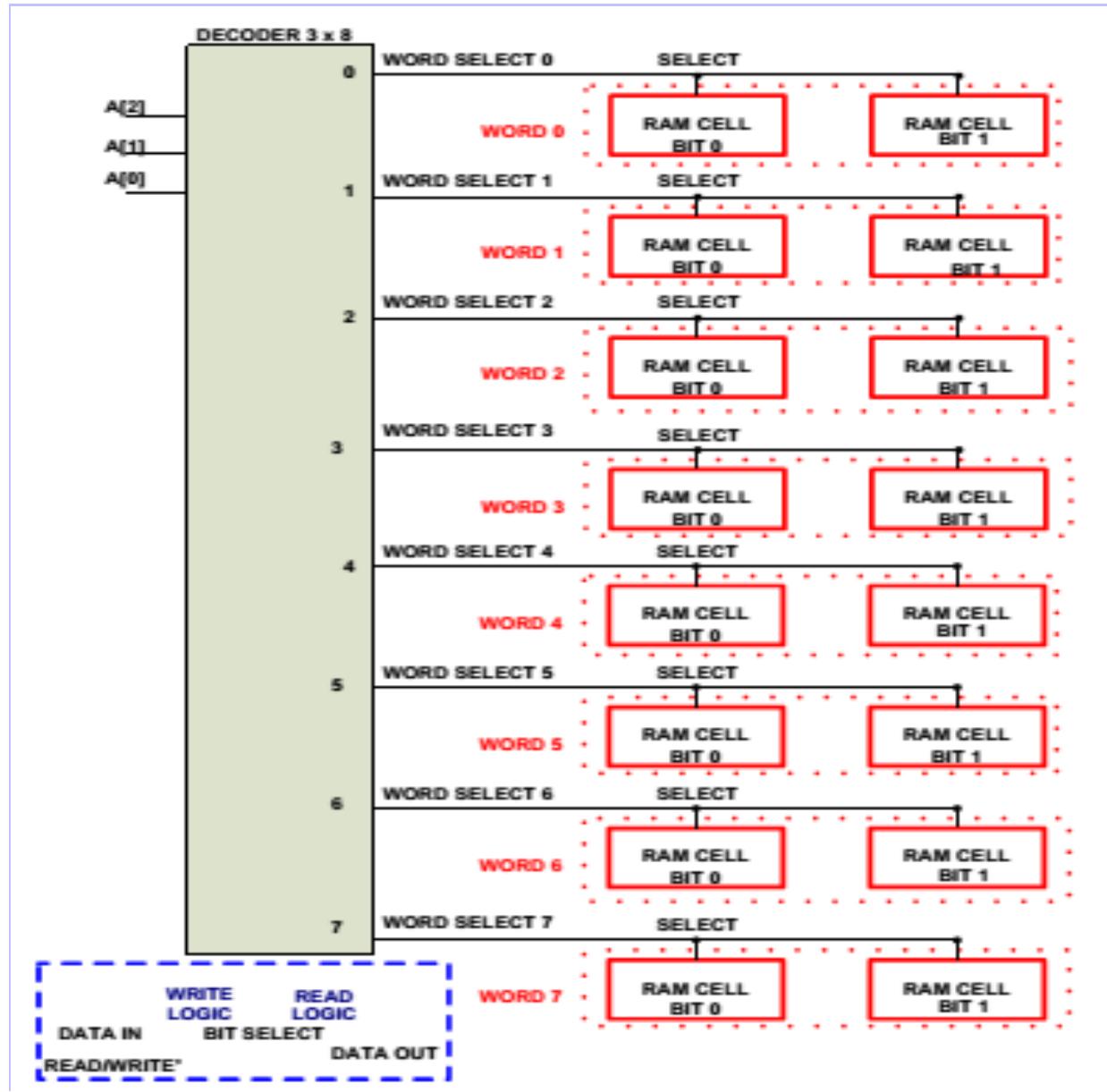
Memory Specification

Specifications



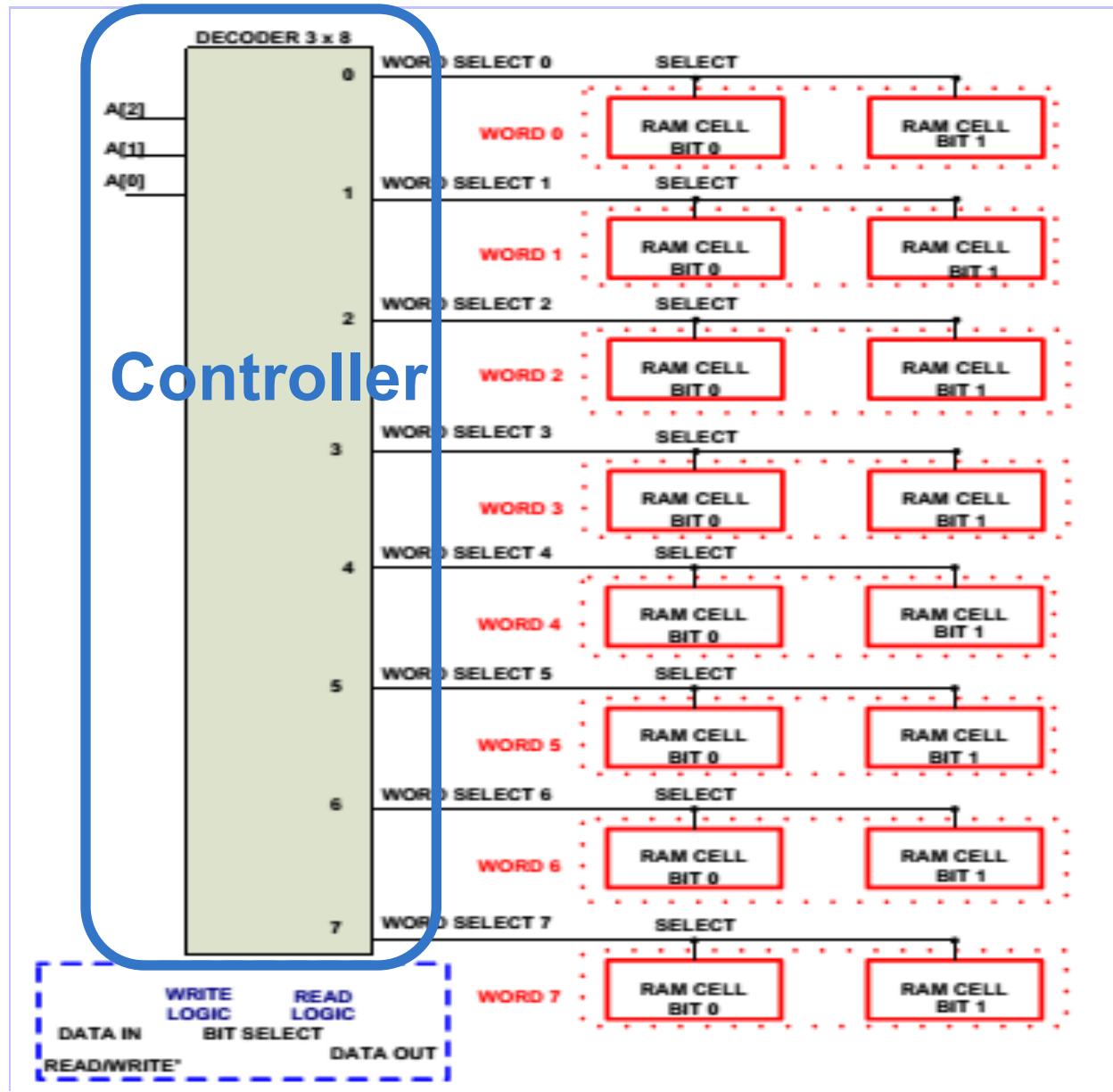
Memory Specification

Specifications



Memory Specification

Specifications



ALU Specification

Specifications

BYTE-ORIENTED FILE REGISTER OPERATIONS

		BYTE-ORIENTED FILE REGISTER OPERATIONS							
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	1fff	ffff	Z	2
CLRW	-	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	0011	dfff	ffff	Z	1,2,3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff	Z	1,2,3
IOWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	1fff	ffff		
NOP	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2



ALU Specification

Specifications

BIT-ORIENTED FILE REGISTER OPERATIONS

BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3

LITERAL AND CONTROL OPERATIONS

ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	TO,PD	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into Standby mode	1	00	0000	0110	0011	TO,PD	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

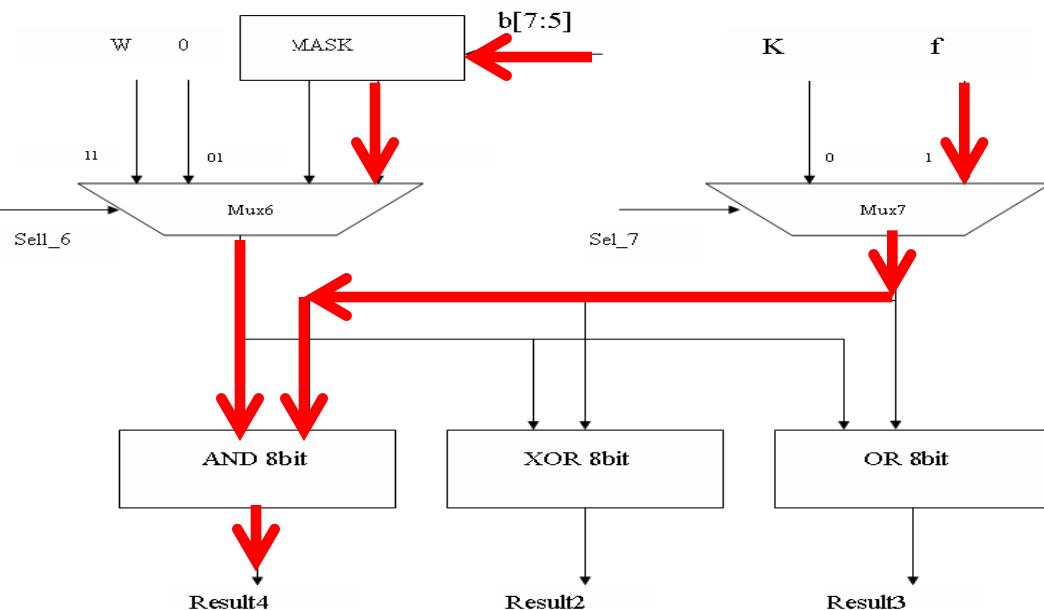
Core CPU Specification

Specifications

BIT-ORIENTED FILE REGISTER OPERATIONS

BCF	f, b	Bit Clear f
BSF	f, b	Bit Set f
BTFSC	f, b	Bit Test f, Skip if Clear
BTFSS	f, b	Bit Test f, Skip if Set

1	01	00bb	bfff	ffff	1,2
1	01	01bb	bfff	ffff	1,2
1(2)	01	10bb	bfff	ffff	3
1(2)	01	11bb	bfff	ffff	3



Byte-oriented file register operations

13	8	7	6	0
OPCODE	d	f (FILE #)		

d = 0 for destination W
d = 1 for destination f
f = 7-bit file register address

Bit-oriented file register operations

13	10	9	7	6	0
OPCODE	b (BIT #)	f (FILE #)			

b = 3-bit bit address
f = 7-bit file register address

Literal and control operations

General

13	8	7	0
OPCODE		k (literal)	

k = 8-bit immediate value

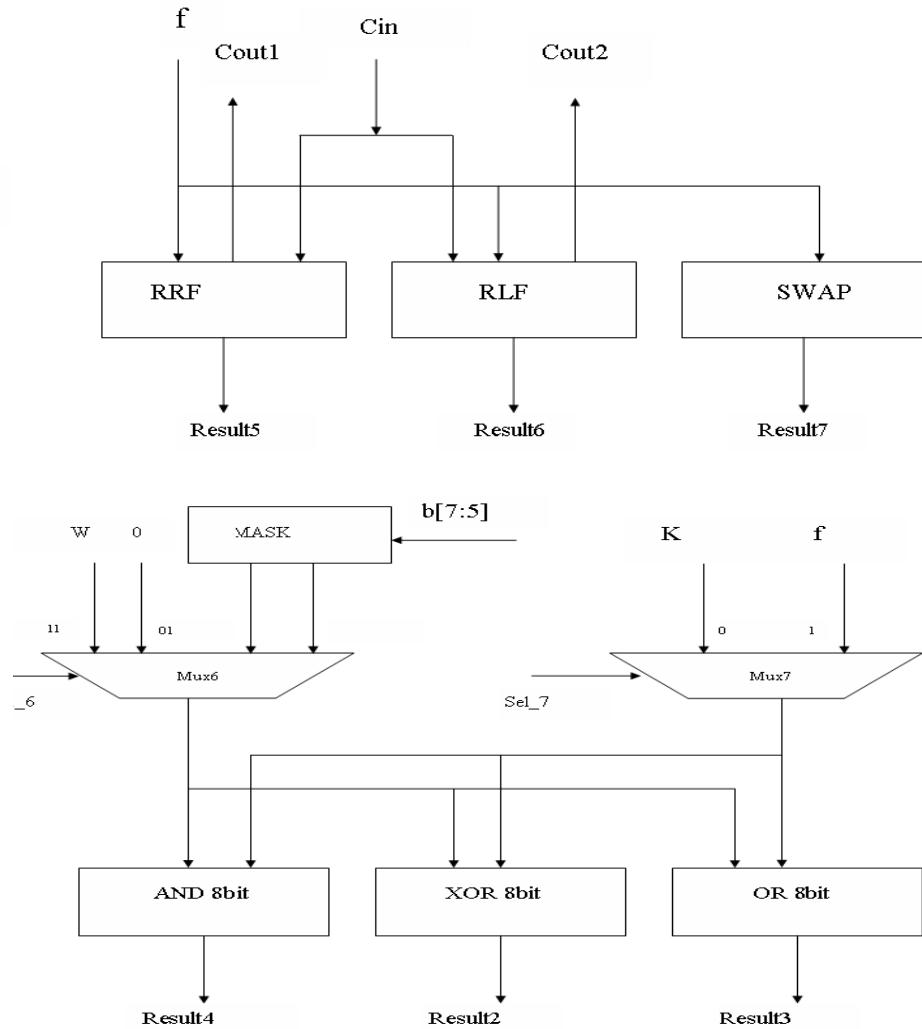
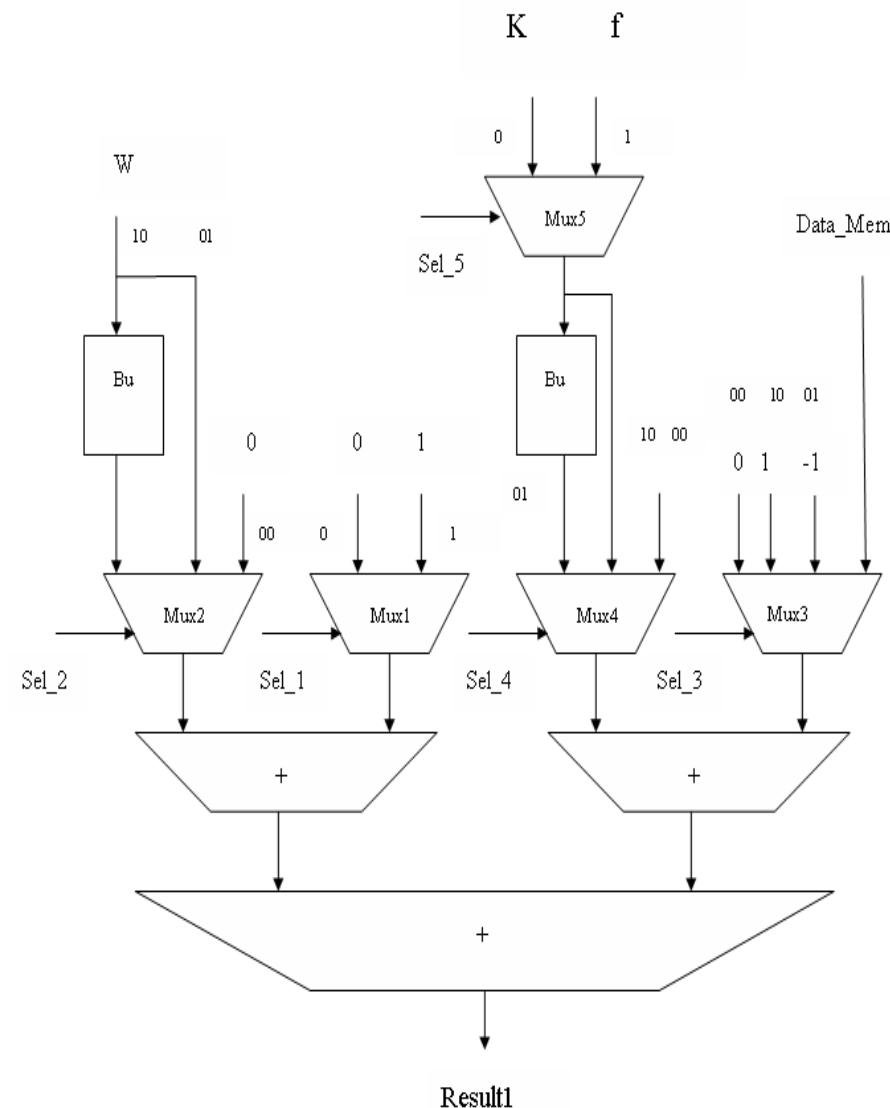
CALL and GOTO instructions only

13	11	10	0
OPCODE		k (literal)	

k = 11-bit immediate value

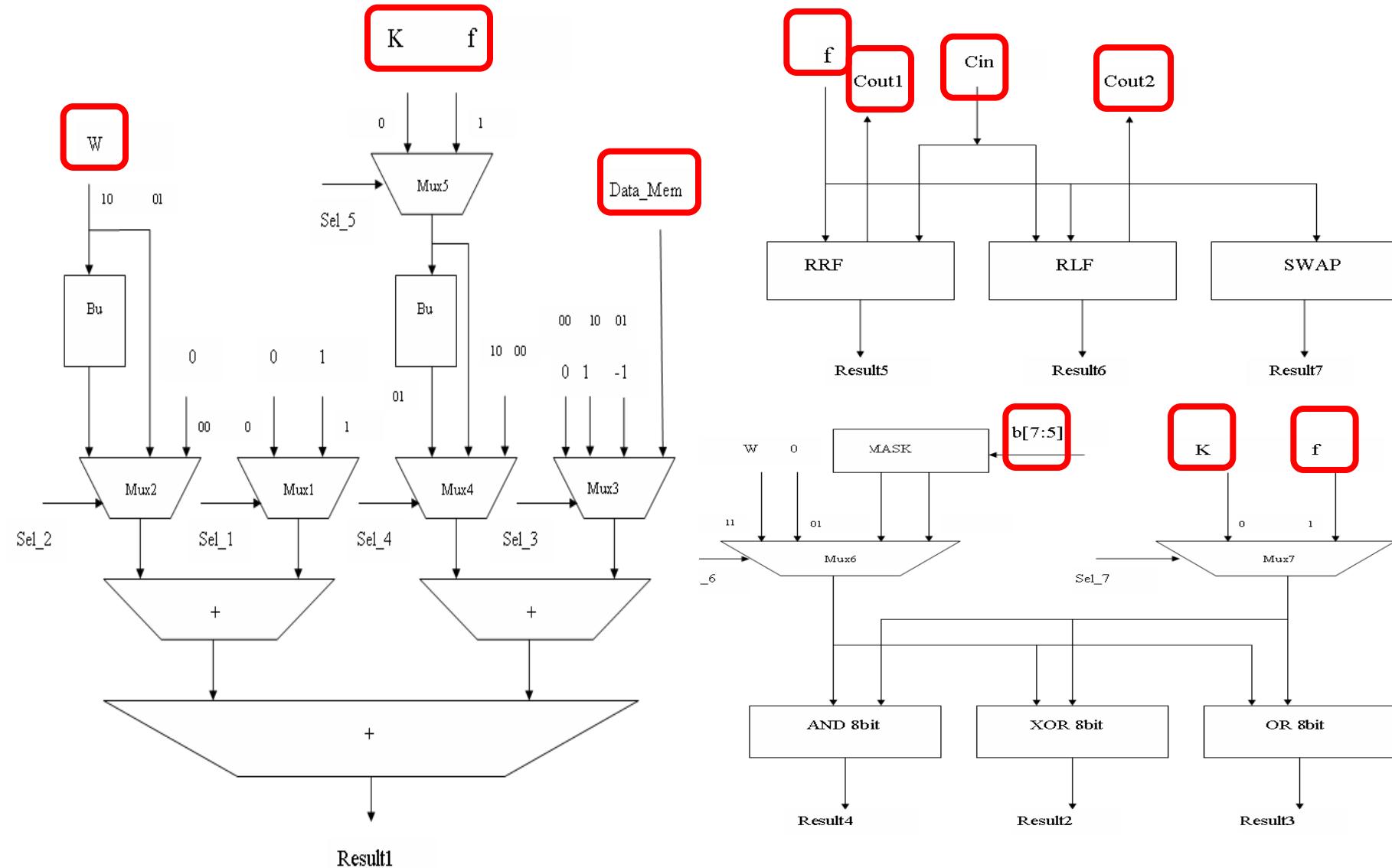
ALU Specification

Specifications



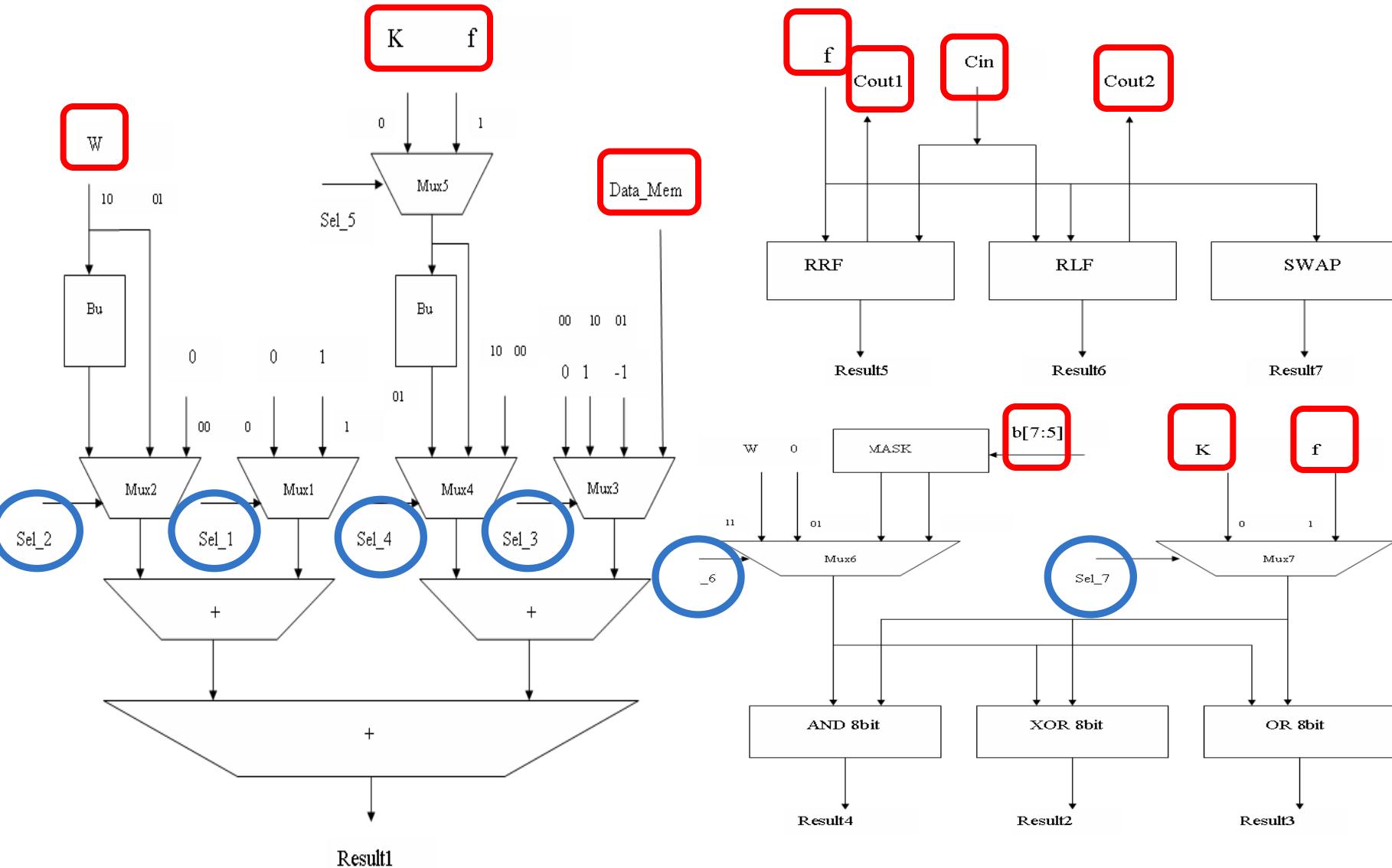
ALU Specification

Specifications



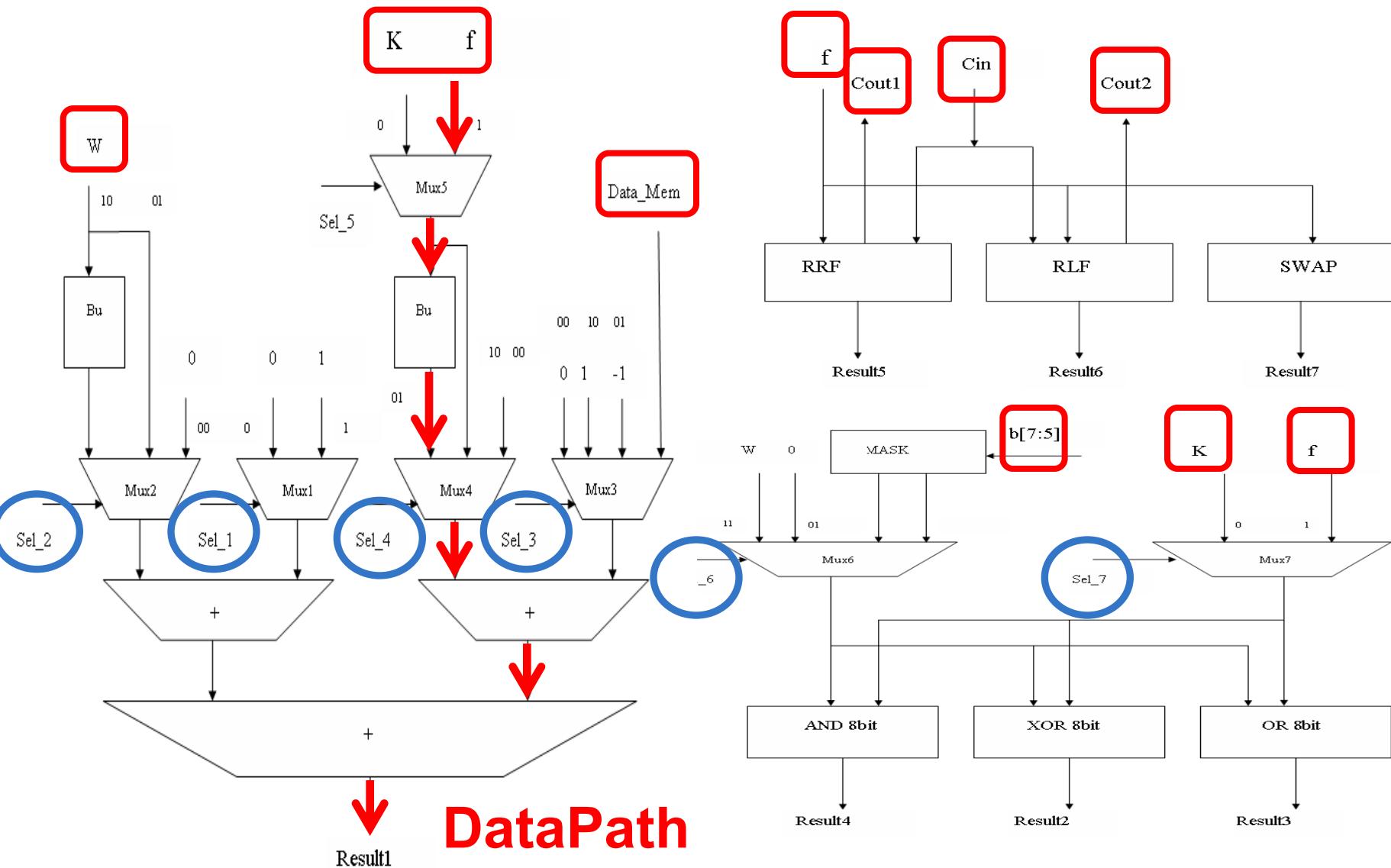
ALU Specification

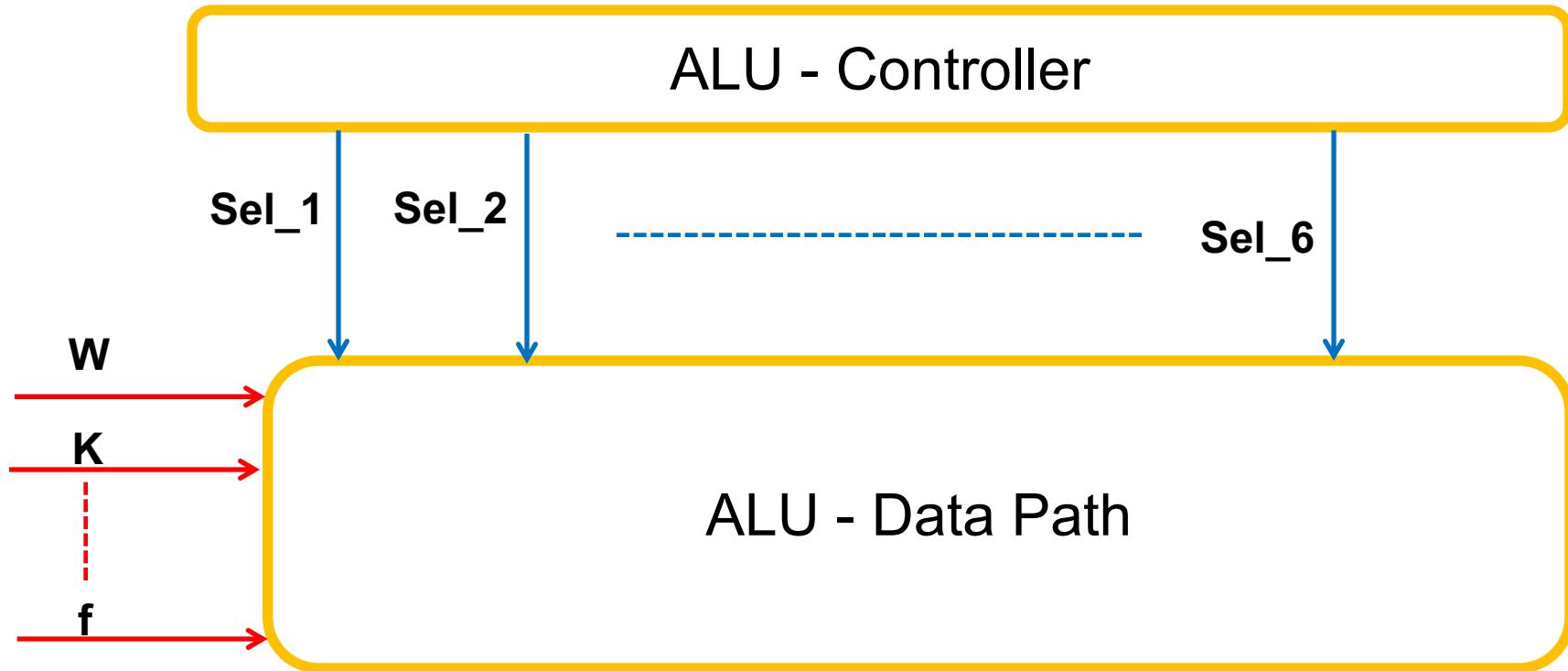
Specifications



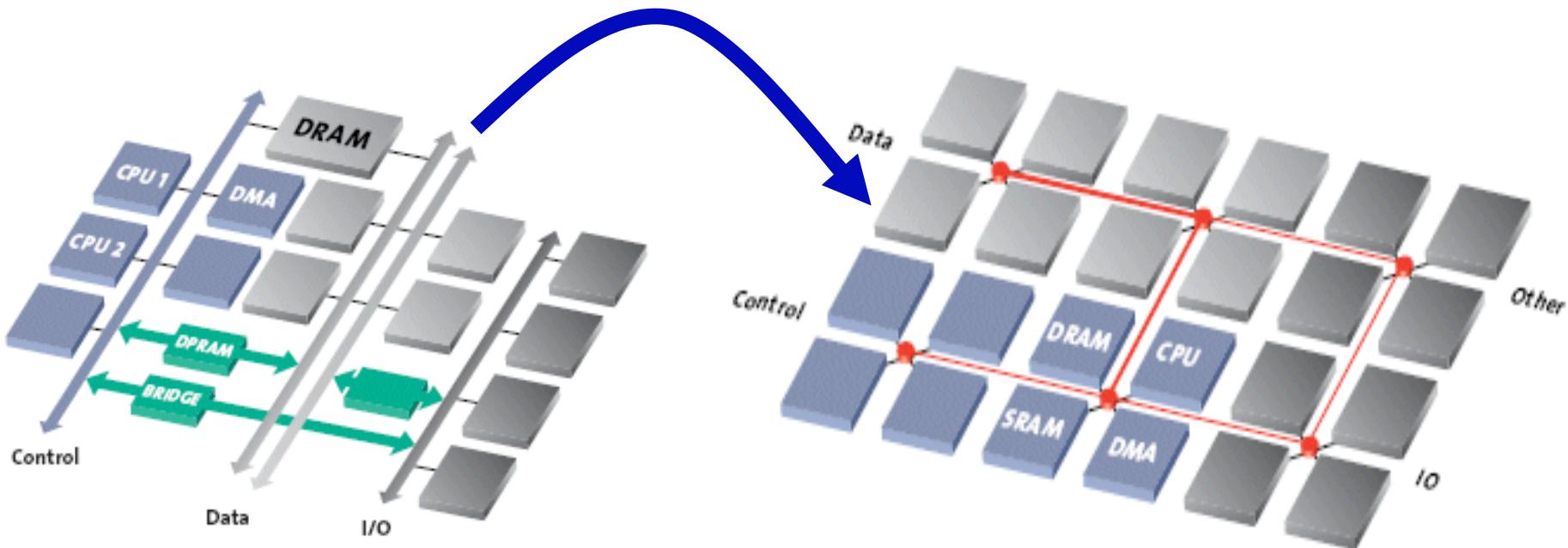
ALU Specification

Specifications

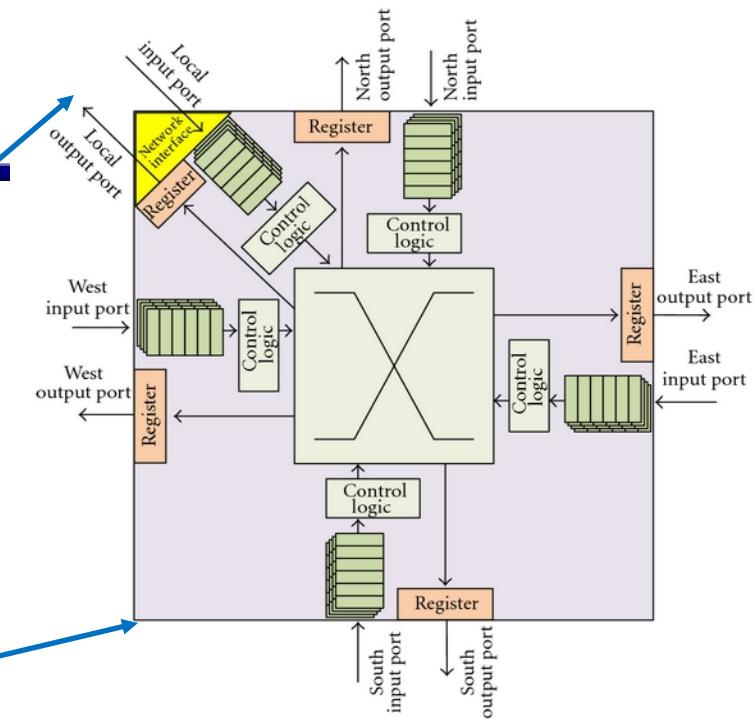
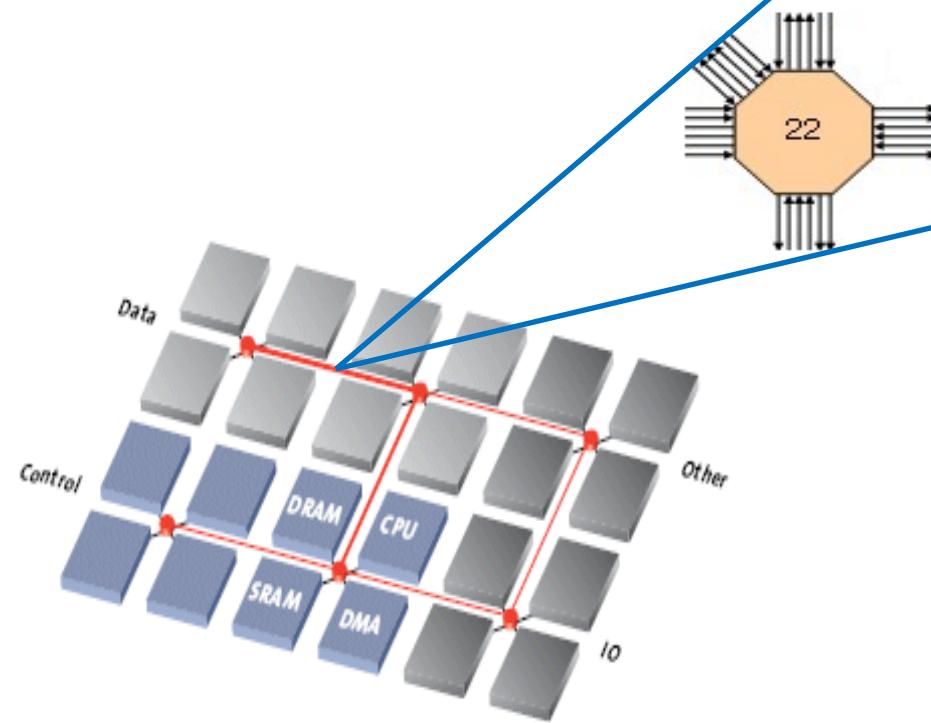




Network-On-Chip (NoC) has been proposed as replacing traditional bus architectures to solve complexity issues such as synchronous frequency, infrastructure interconnect and connective protocols when integrating hundreds IP core or systems



NoC Specification



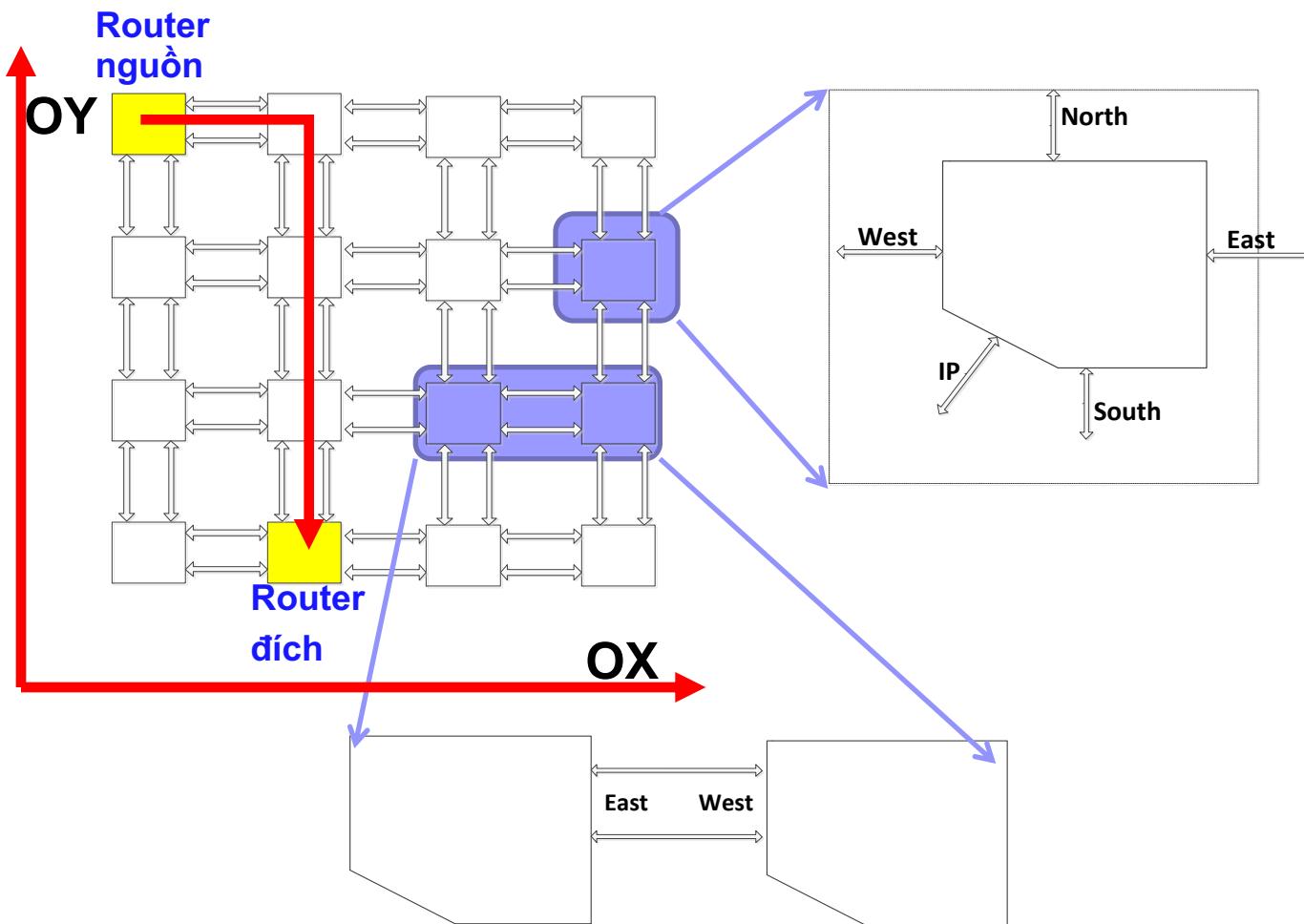
Header
flit
Body flit

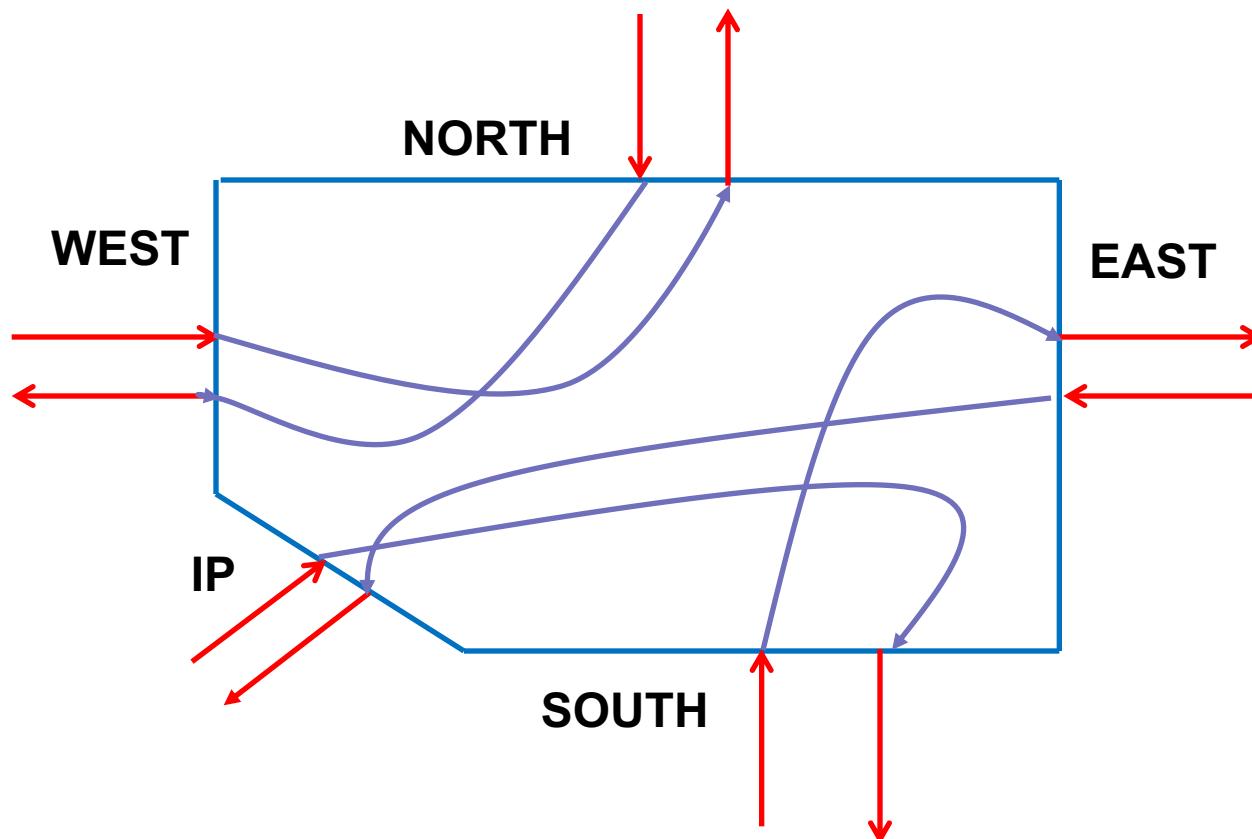
BW	RC	VA	SA	ST	LT
BW	RC	VA	SA	ST	LT

- **BW:** Buffer write
- **RC:** Router computation
- **VA:** Virtual allocation
- **SA:** Switch allocation
- **ST:** Switch traversal
- **LT :** Link traversal

Amit Kumar Li-Shiuan Peh, Parth Kundu, Niraj K.Jha, "Toward Ideal On-Chip Communication Using Express Virtual Channels", micr-28-01-kuma.3d.IEEE 2008, Princeton University.

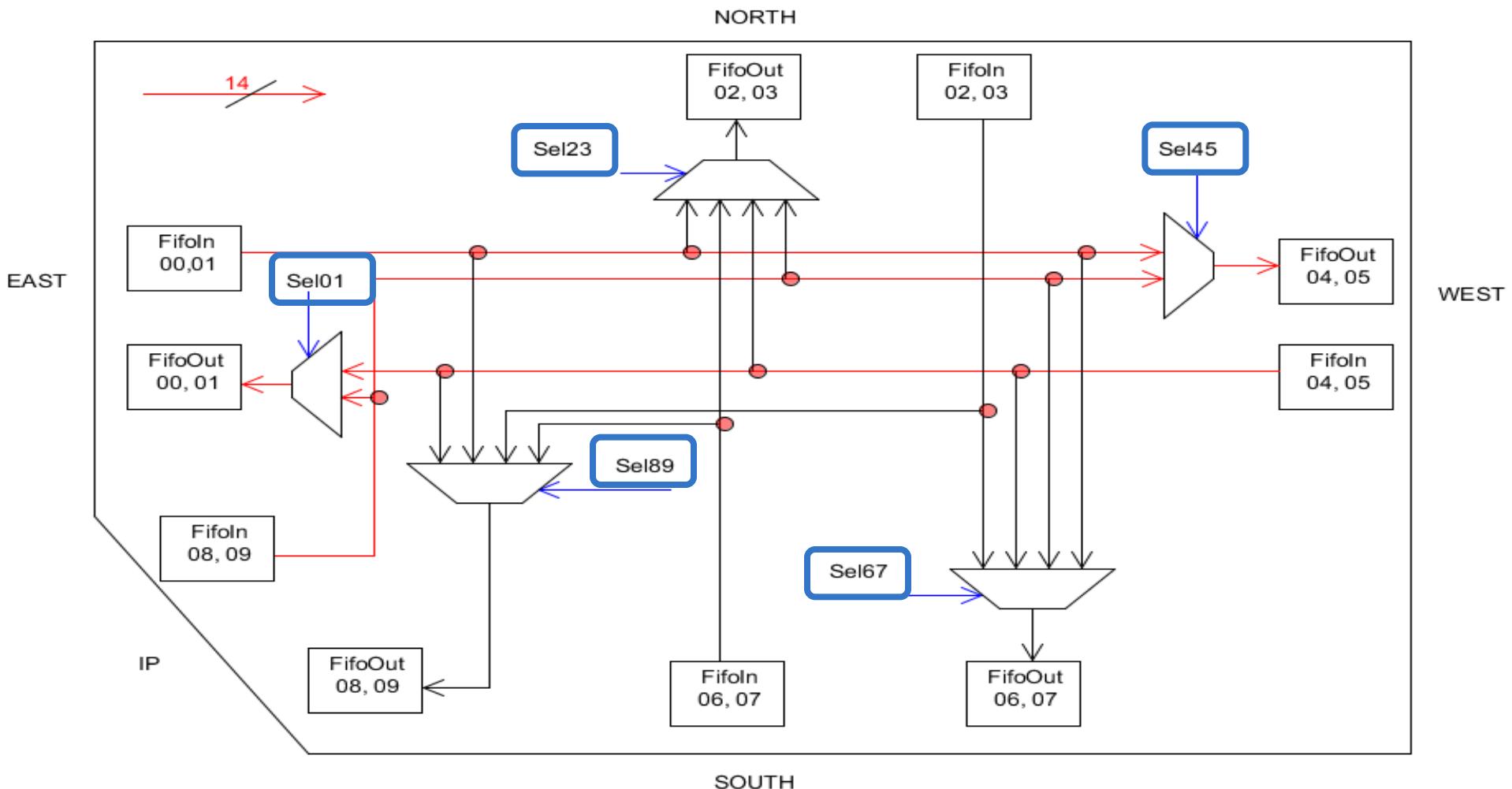
Mô Hình NoC 4x4





NoC Specification

Specifications



How to start specification ?

Specifications

DataPath

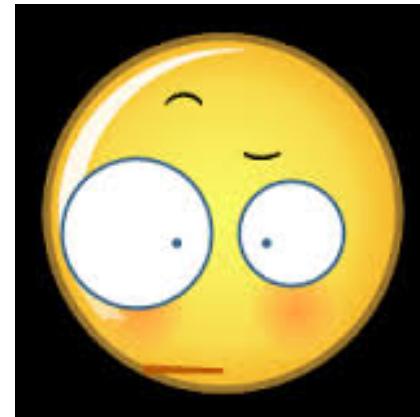
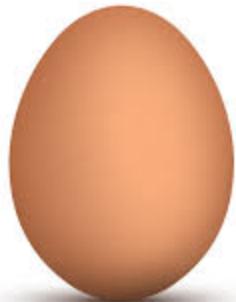
Controller



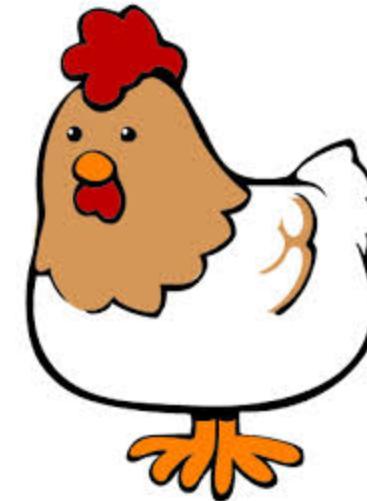
How to start specification ?

Specifications

DataPath



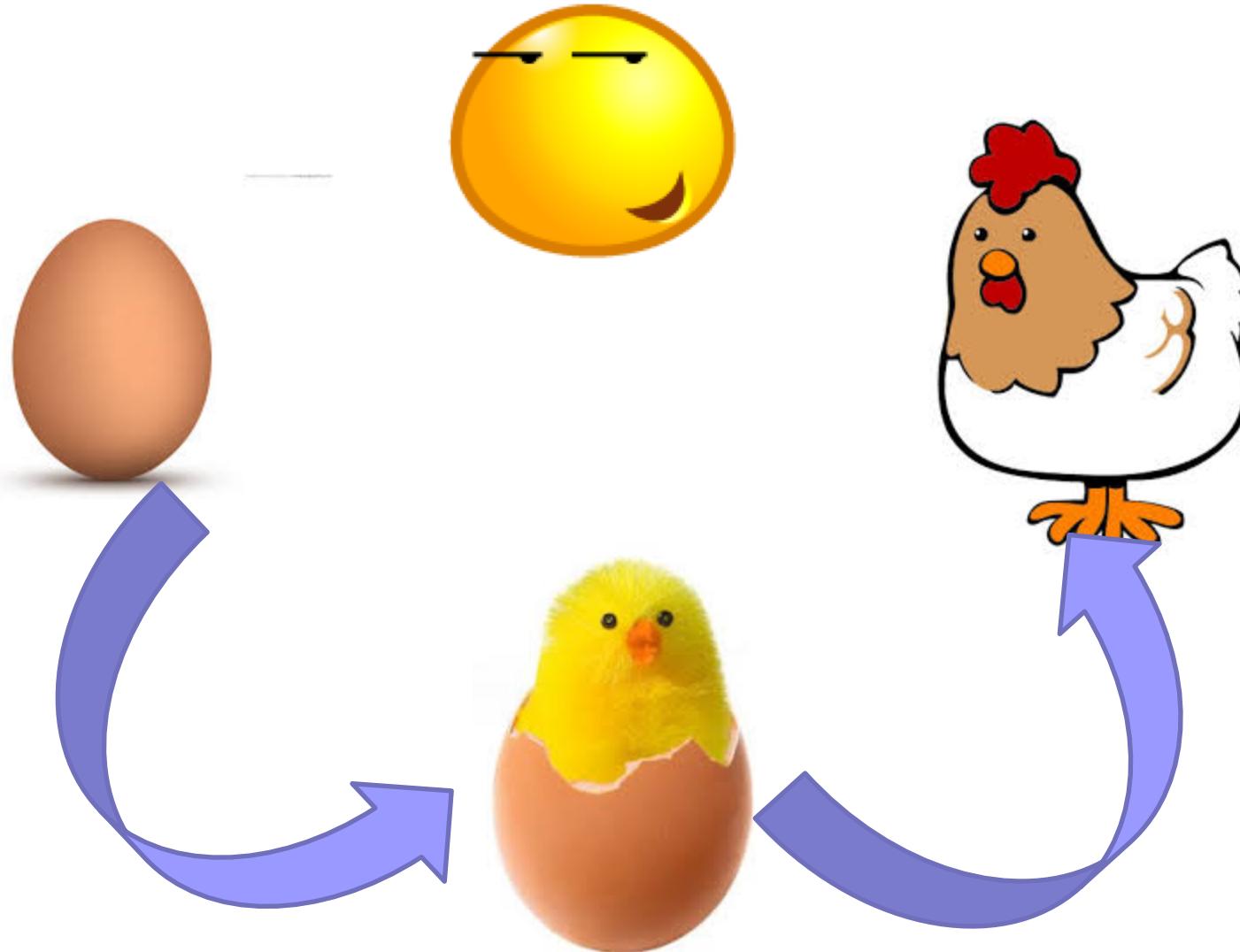
Controller



How to start specification ?

Specifications

DataPath



Controller



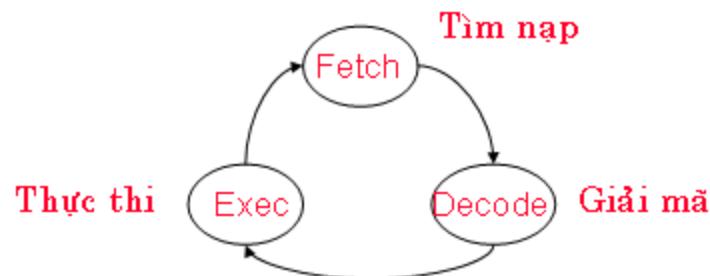
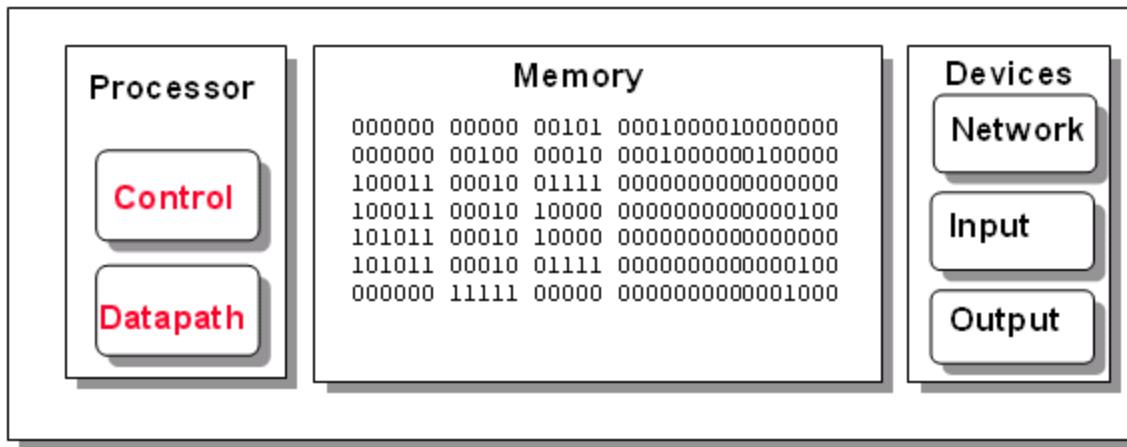
Cell Based Design & Specification

- ❖ Cell Based Design Flow
- ❖ How To Start A Specification
- ❖ Specification Examples
- ❖ Controller and Data Hazard

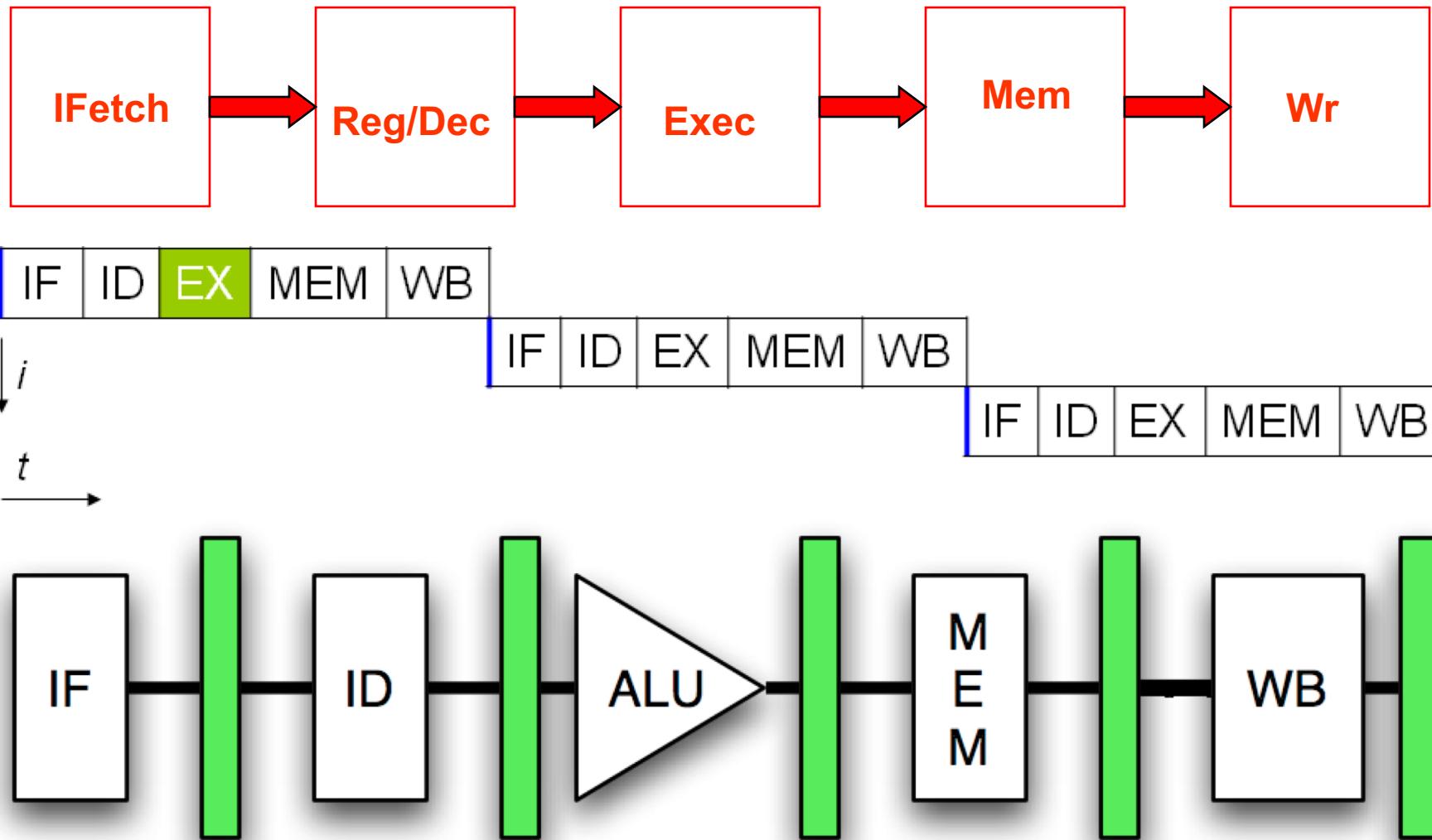
Controller

- Pipeline Concept**
- Pipe line not using state machine**
- Pipeline with state machine**

CPU & Processing

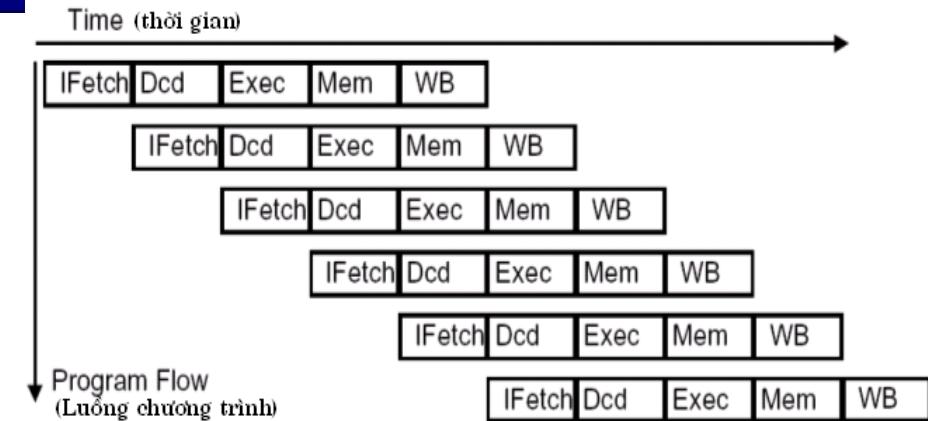
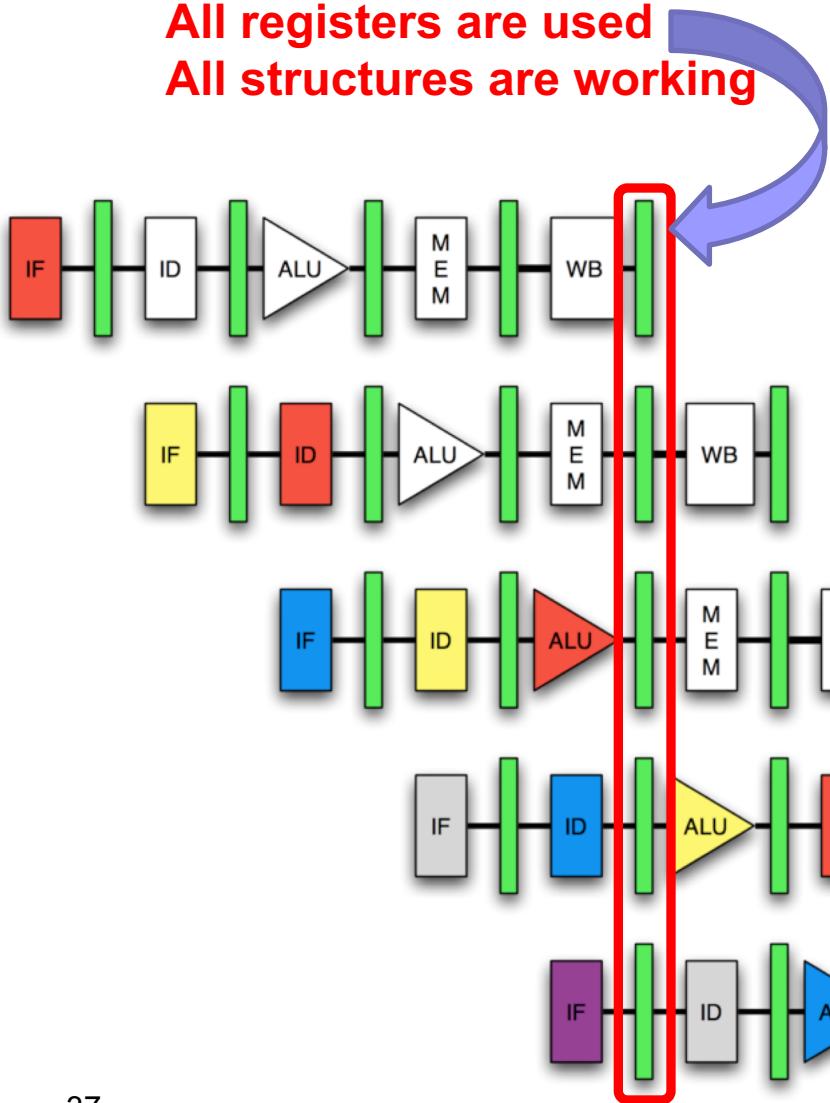


Processing



Pipeline Process

All registers are used
All structures are working

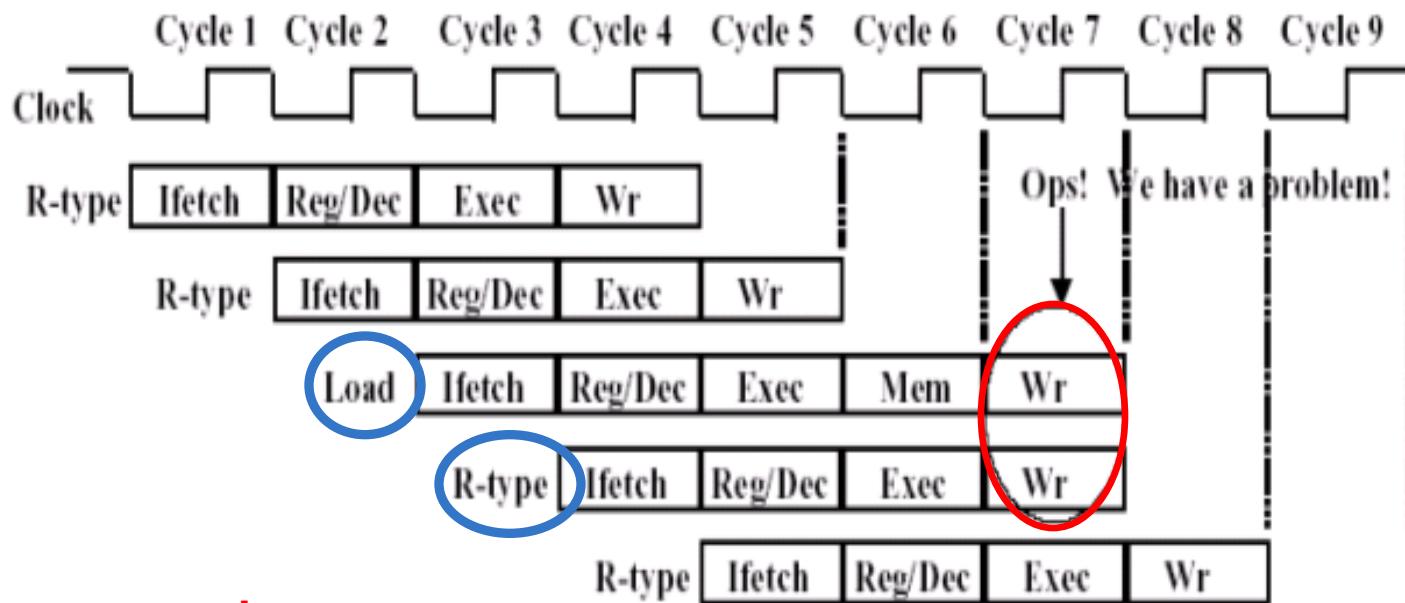




Hazard & Solution

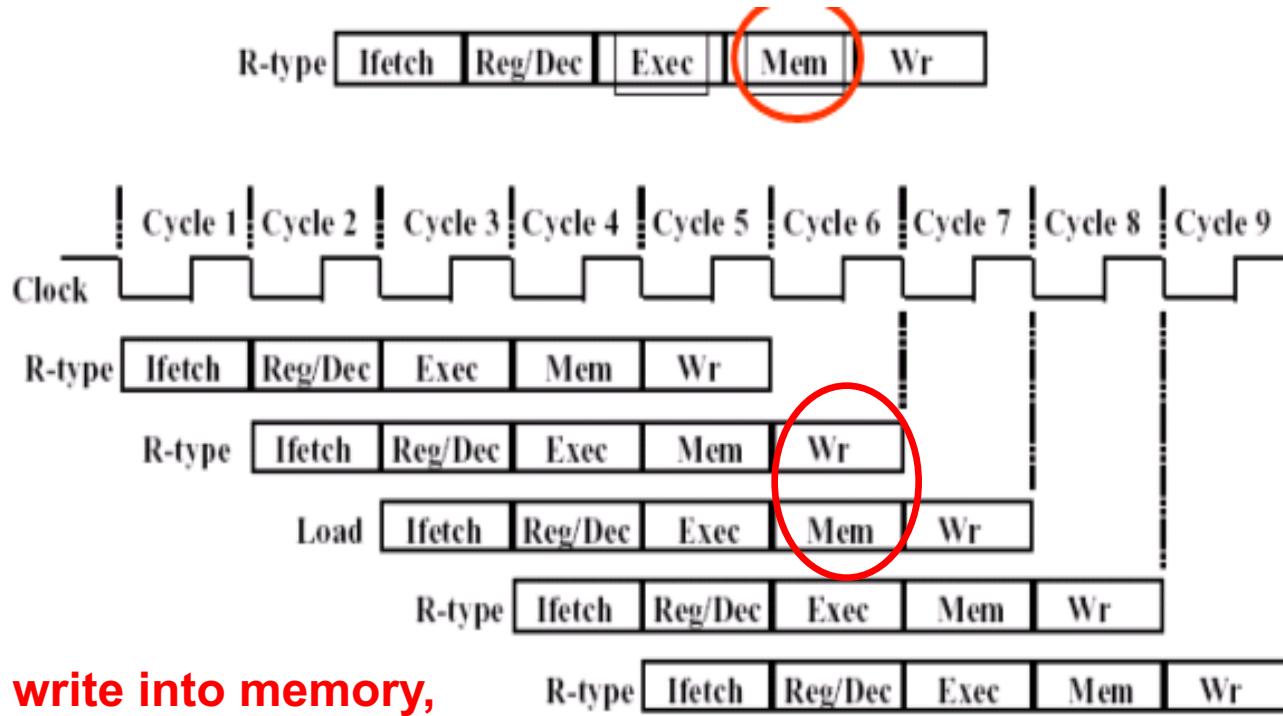
- Structure Hazard
- Data Hazard

Structure Hazard



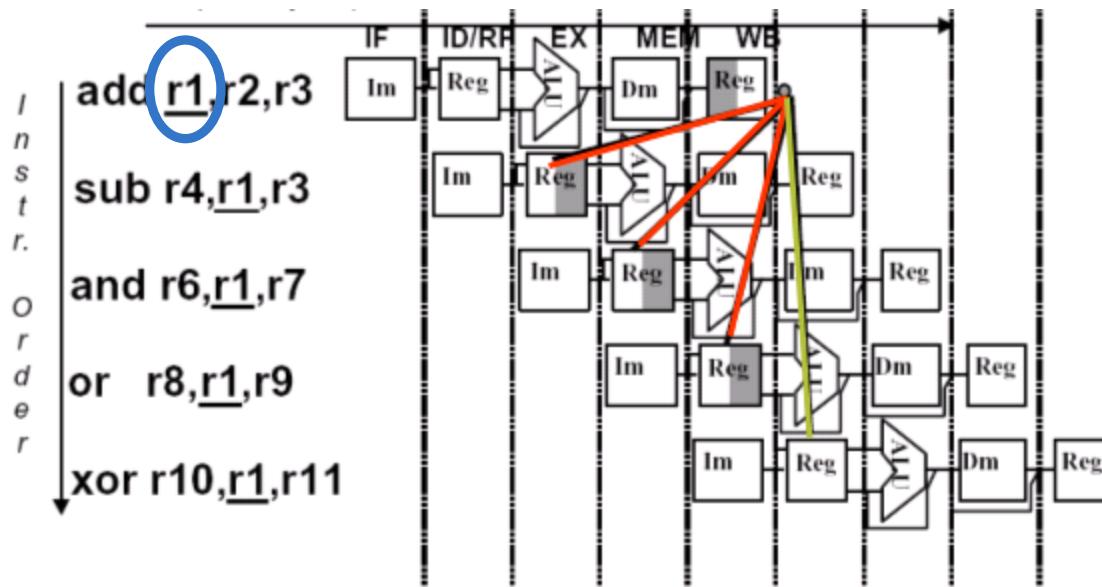
Some commands as
'R' type only need 4 states,
But some command as 'Load' type need 5 states

Structure Hazard



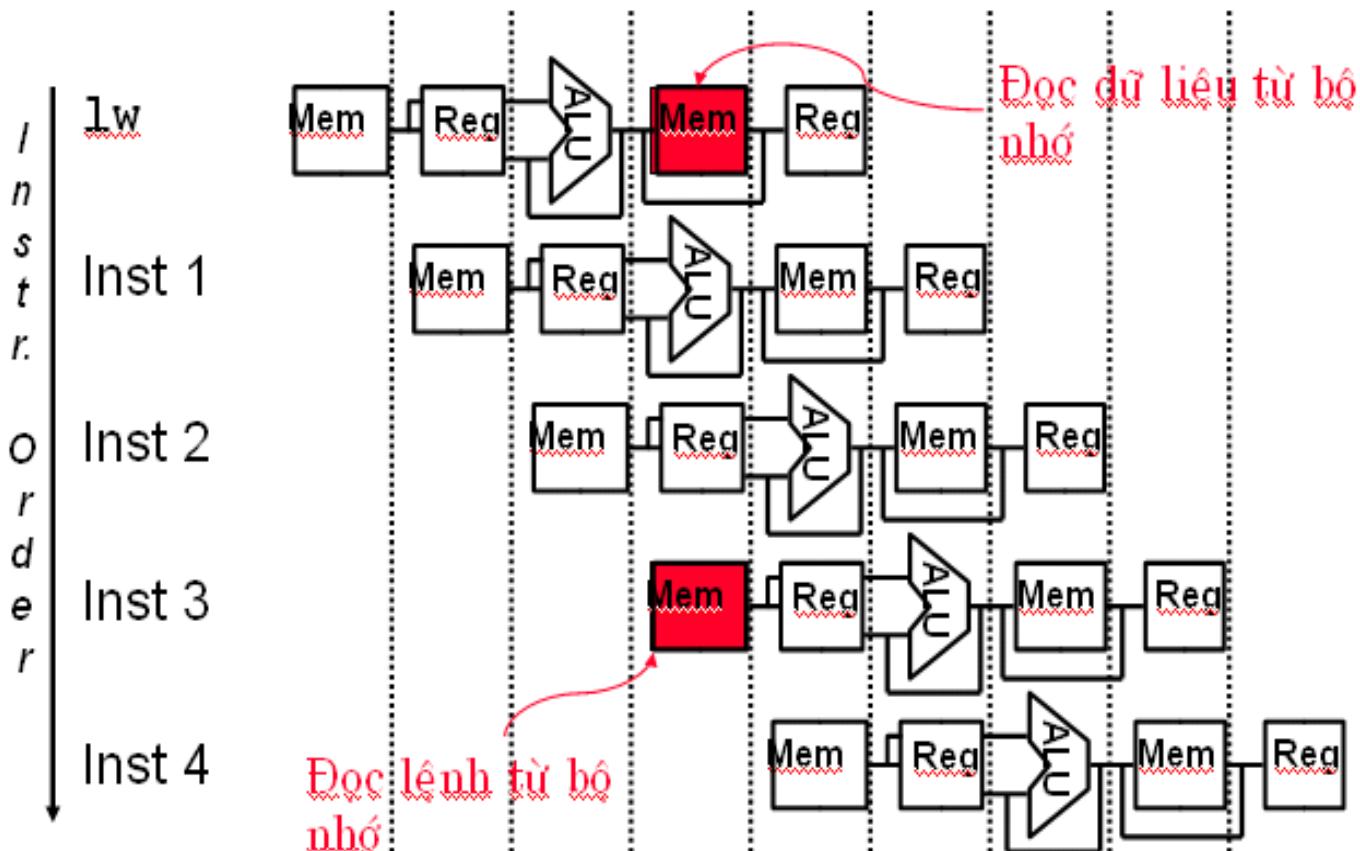
**Data need to write into memory,
but such memory are being in work**

Structure Hazard



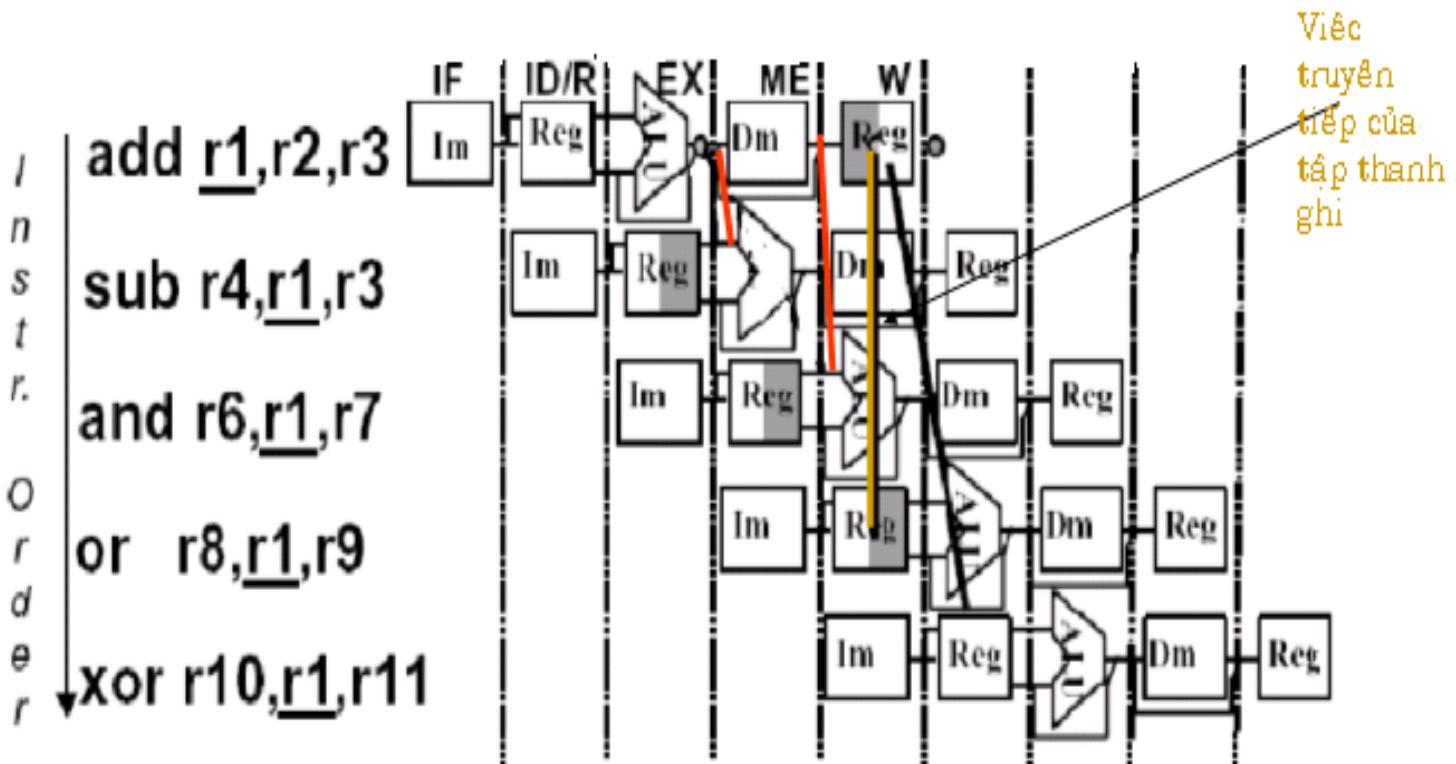
**Register is used next commands,
while it has not updated in previous command**

Structure Hazard



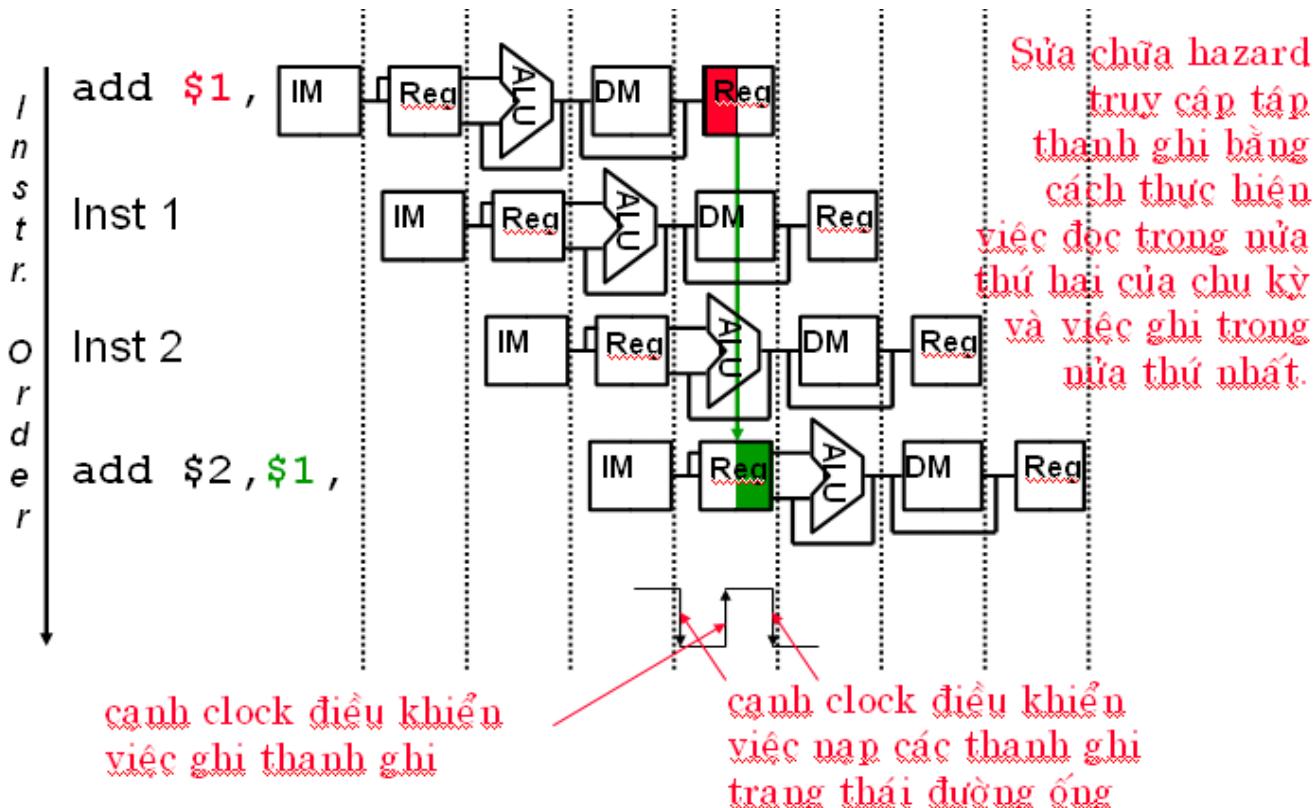
Read data from memory, while it has not finished updating

Structure Hazard



→ **Solution:** Do not need to updated register,
the ready value is transferred to next command

Structure Hazard



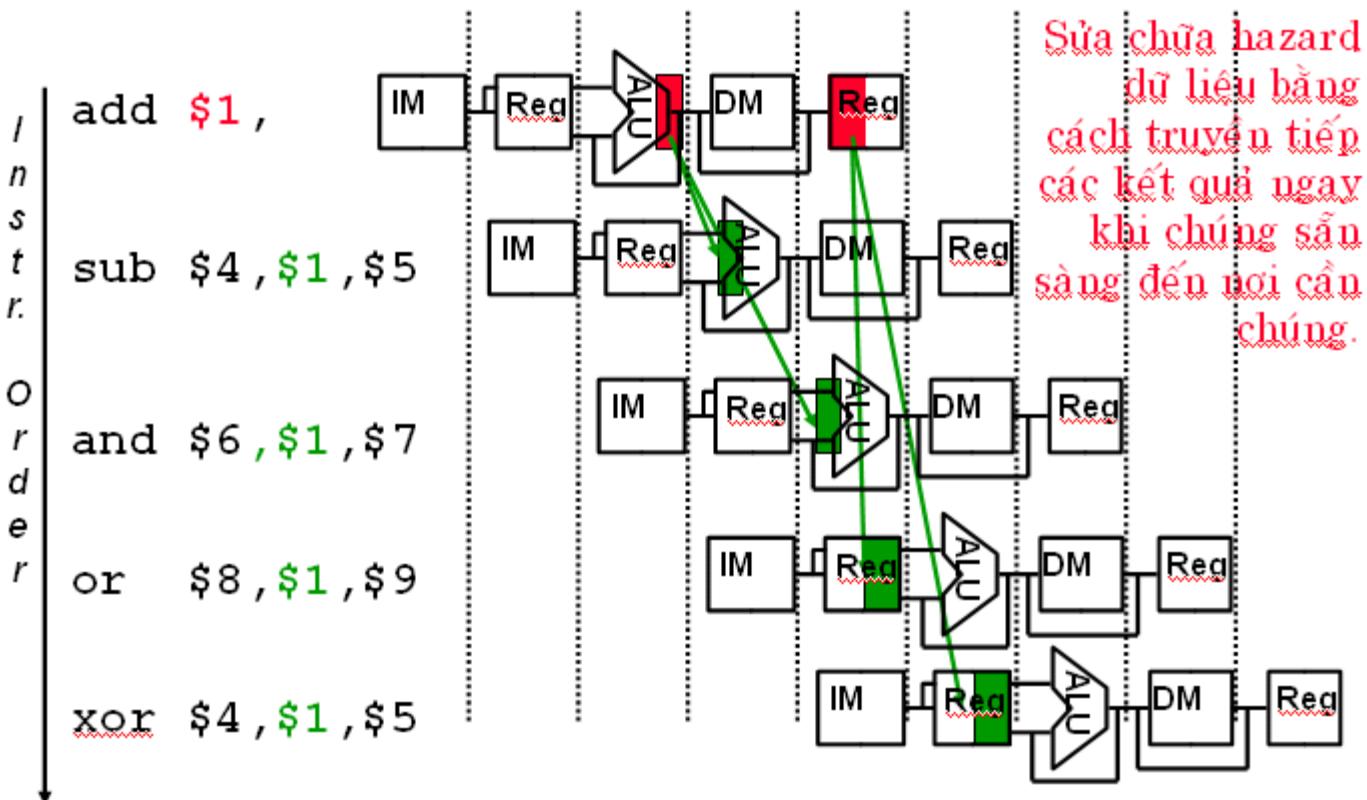
→ Solution: Read or write data based on different edges of clock



Hazard & Solution

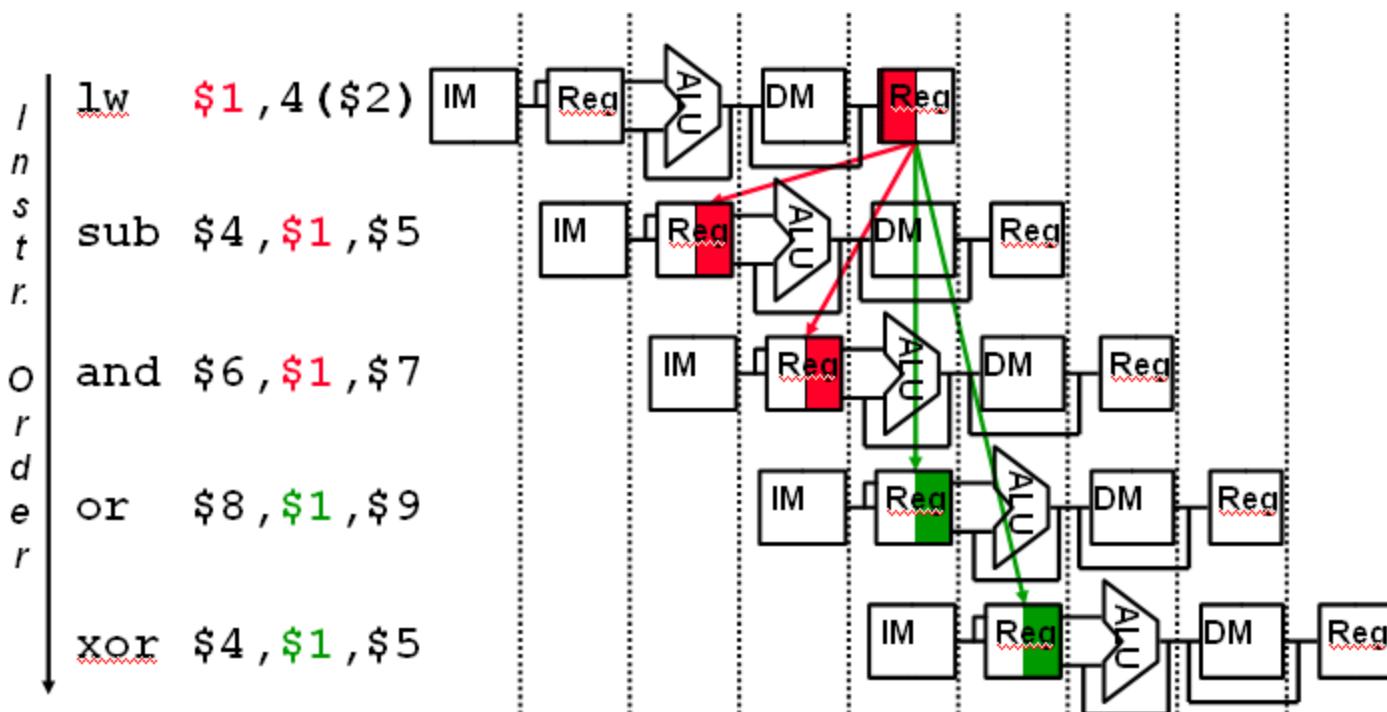
- Structure Hazard
- Data Hazard

Data Hazard



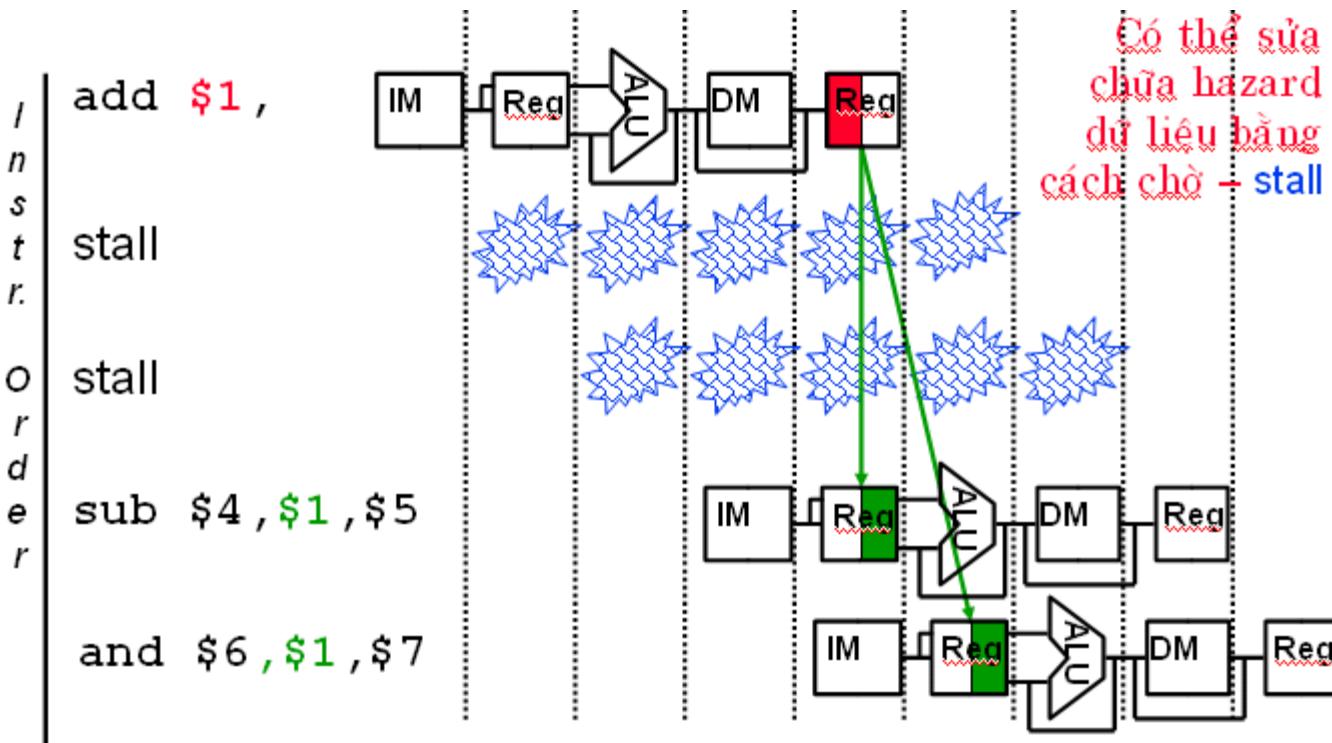
→ Solution: Used ready data immediately

Data Hazard



Cannot apply ready data due to long delay

Data Hazard



→ Solution: NOP command is inserted to delay some states