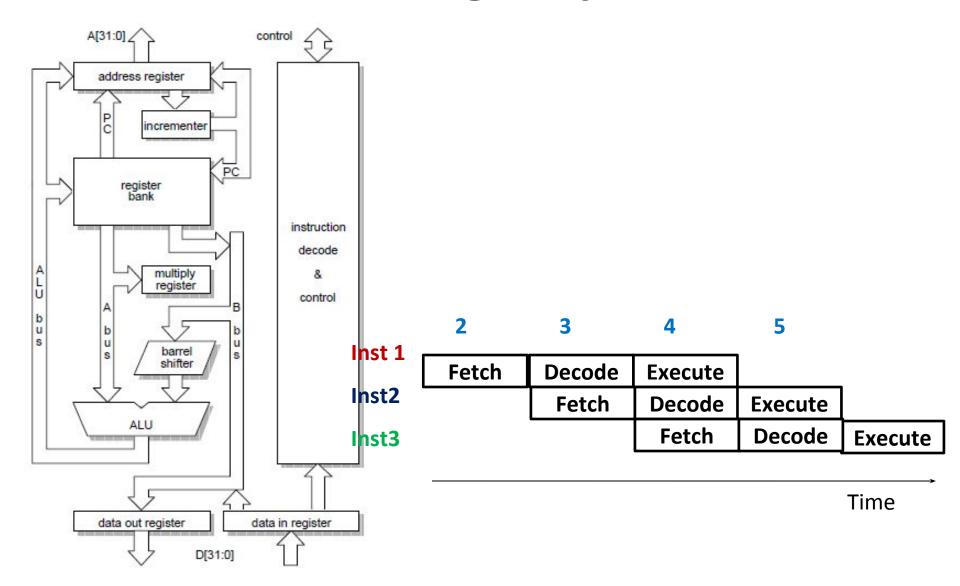
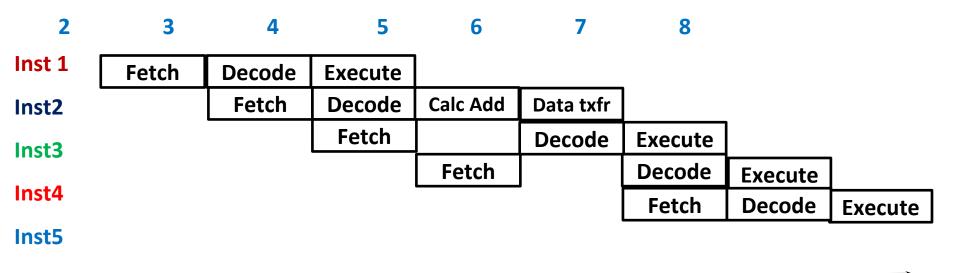
## **Embedded Systems**

Dr. Sajesh Kumar U.

## **ARM 3 Stage Pipeline**

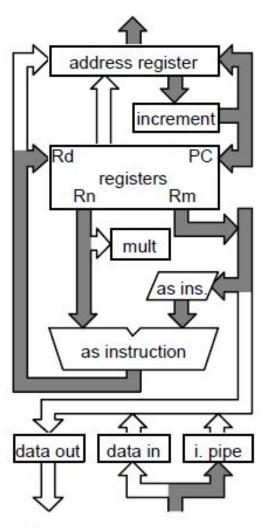


# Multi cycle instruction in 3 stage pipeline

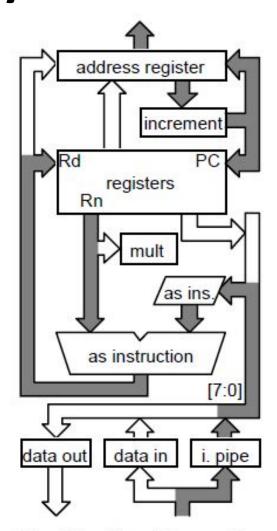


Time

# Data Processing Instruction-Data path Activity

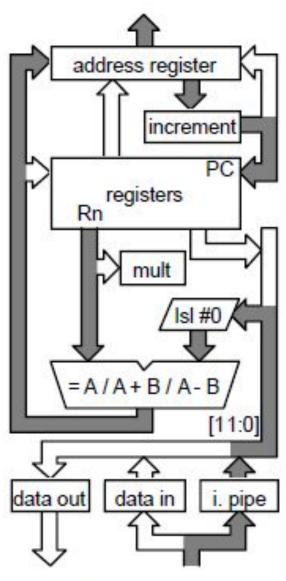


(a) register - register operations

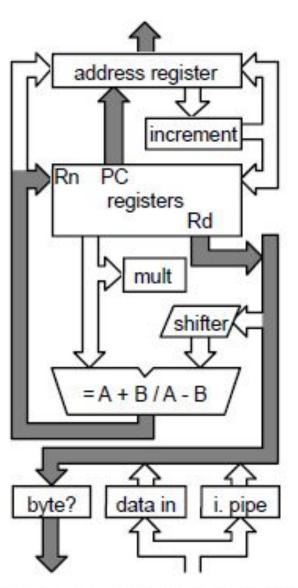


(b) register - immediate operations

#### Store Instruction- Data path activity

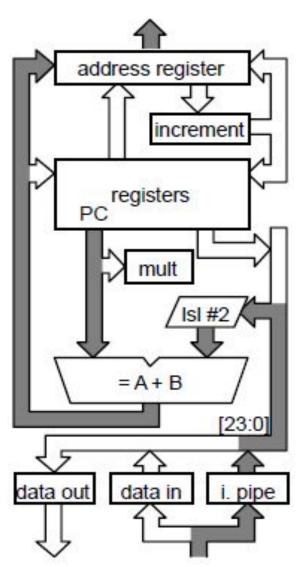


(a) 1st cycle - compute address



(b) 2nd cycle - store data & auto-index

# Branch Instruction –Data path activity



address register increment R14 registers mult shifter = A data out data in i. pipe

(b) 2nd cycle - save return address

(a) 1st cycle - compute branch target

## **5 Stage Pipeline**

	Clock cycle number										
instr	1	2	3	4	5	6	7	8			
load	IF	ID	EX	MEM	WB						
Instr 1		IF	ID	EX	MEM	WB					
Instr 2			IF	ID	EX	MEM	WB				
Instr 3				IF	ID	EX	MEM	WB			

## **ARM 5 stage Pipeline Architecture**

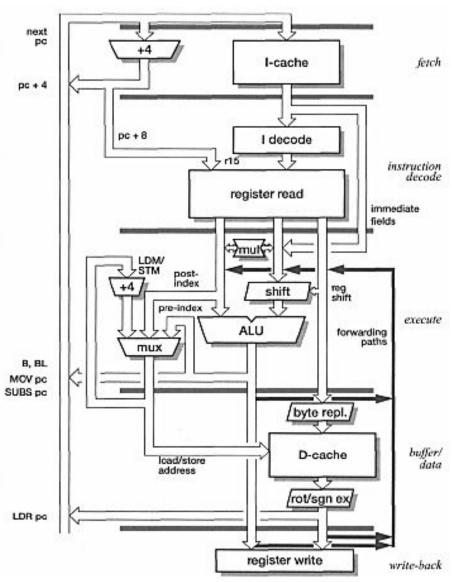


Figure 4.4 ARM9TDMI 5-stage pipeline organization.

# 5 Stage Pipeline Execution and Instruction stalling-Structural Hazards

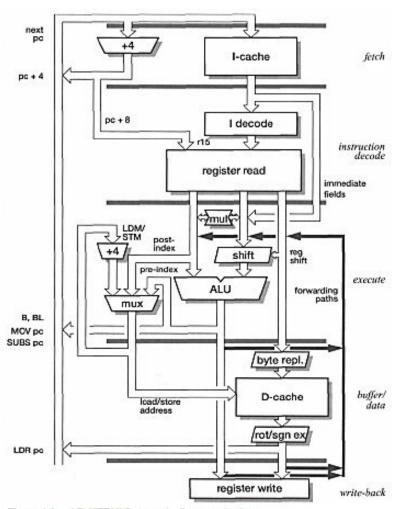
	Clock cycle number										
instr	1	2	3	4	5	6	7	8			
load	IF	ID	EX	MEM	WB						
Instr 1		IF	ID	EX	MEM	WB					
Instr 2			IF	ID	EX	MEM	WB				
Instr 3				IF	ID	EX	MEM	WB			

	Clock cycle number											
instr	1	2	3	4	5	6	7	8	9			
load	IF	ID	EX	MEM	WB							
Instr 1		IF	ID	EX	MEM	WB						
Instr 2			IF	ID	EX	MEM	WB					
Instr 3	15			stall	IF	ID	EX	MEM	WB			

#### **Data Hazards**

			(	Clock c	ycle nu	ımber				
		1	2	3	4	5	6	7	8	9
ADD	R1,R2,R3	IF	ID	EX	MEM	WB				
SUB	R4,R5,R1		IF	ID <sub>sub</sub>	EX	MEM	WB			
AND	R6,R1,R7			IF	ID <sub>and</sub>	EX	MEM	WB		
OR	R8,R1,R9				IF	ID <sub>or</sub>	EX	MEM	WB	
XOR	R10,R1,R11					IF	ID <sub>xor</sub>	EX	MEM	WB

## Data forwarding mechanism



	Clock cycle number										
		1	2	3	4	5	6	7			
ADD	R1,R2,R3	IF	ID	EX -	MEM-	WB					
SUB	R4,R5,R1		IF	ID <sub>sub</sub>	ĚΧ	MEM	WB				
AND	R6,R1,R7				ID <sub>and</sub>	EX	MEM	WB			

			(	Clock	cycle n	umber				
		1	2	3	4	5	6	7	8	9
ADD	R1,R2,R3	IF	ID	EX	MEM	WB=	<b>7</b> /			
SUB	R4,R5,R1		IF	stall	stall	ID <sub>sub</sub>	EX	MEM	WB	
AND	R6,R1,R7		8	stall	stall	IF	ID <sub>and</sub>	EX	MEM	WB

Figure 4.4 ARM9TDMI 5-stage pipeline organization.

### **Memory Access and instruction Stall**

		1	2	3	4	5	6	7	8	9
LDR	R1,@(R2)	IF	ID	EX	MEM-	WB =	1			
SUB	R4,R1,R5		IF	ID	stall	EX <sub>sub</sub>	MEM	WB		
AND	R6,R1,R7			IF	stall	ID	ĒΧ	MEM	WB	
OR	R8,R1,R9				stall	IF	ID	ĒΧ	MEM	WB

#### **Load Interlock**

Cycle			1	2	3	4	5	6	7	8	9
Operation	1		1		:			;		1	:
ADD	R1, R1, R2	F D	E		w					:	
SUB	R3, R4, R1	F	D	E		W				7	
LDR	R4, [R7]		F	D	E	M	W			1	1
ORR	R8, R3, R4			F	D	1	E		W	1	
AND	R6, R3, R1		7		F	1	D	E		W	
EOR	R3, R1, R2		1	Ī			F	D	E		W

	F - Fetch	D - De		E - Es W - Wri			terlock	M -	Memoi	y.		
Cycle				1	2	3	4	5	6	7	8	9
Operation												
ADD	R1, R1, R2	F	D	E		W					1	
SUB	R3, R4, R1		F	D	E		W					
LDR	R4, [R7]			F	D	E	M	W			1	
AND	R6, R3, R1				F	D	E	<b>V</b> ,	W	1	1	
ORR	R8, R3, R4			1		F	D	E		W	T	
EOR	R3, R1, R2			7	Ī		F	D	E		W	I

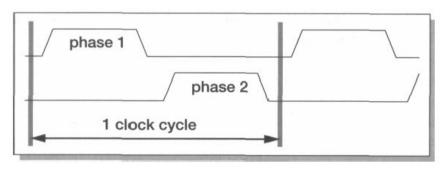
#### LDM Interlock

Cycle				1	2	3	4	5	6	7	8	9	10
Operation													
LDMLA	R13!, {R0-R3}	F	D	E	M	MW	MW	MW	W			1	
SUB	R9, R7, R2		F	D	1	1		E		W		1	
STR	R4, [R9]	**********		F	1	1		D	E	M	W		
ORR	R8, R4, R3		[					F	D	E		W	
AND	R6, R3, R1								F	D	E		W

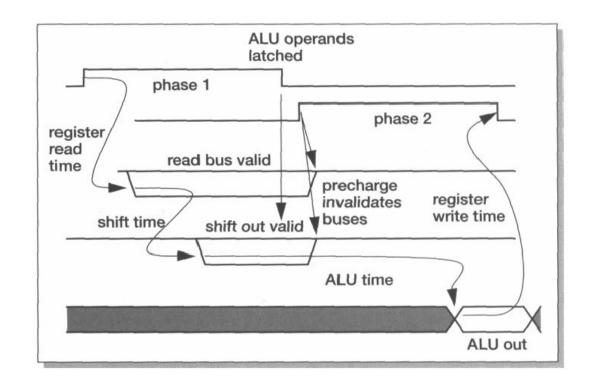
F - Fetch D - Decode E - Excute I - Interlock M - Memory

ME - Simultaneous Memory and Writeback W - Writeback

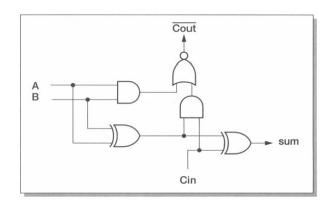
### **ARM Implementation**

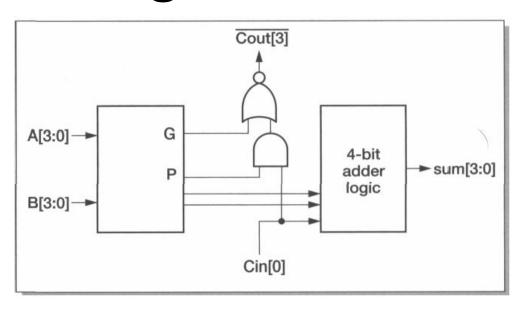


- Register read buses are dynamic
- Two phase non overlapping clock

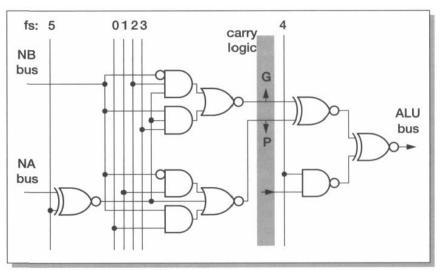


### **Adder Design**

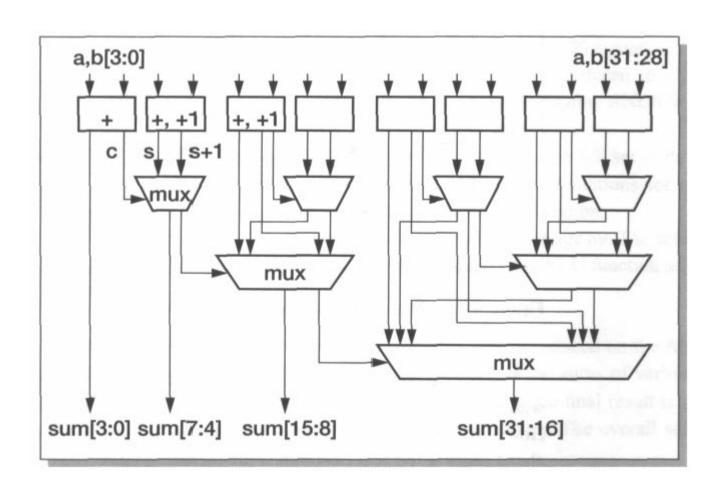




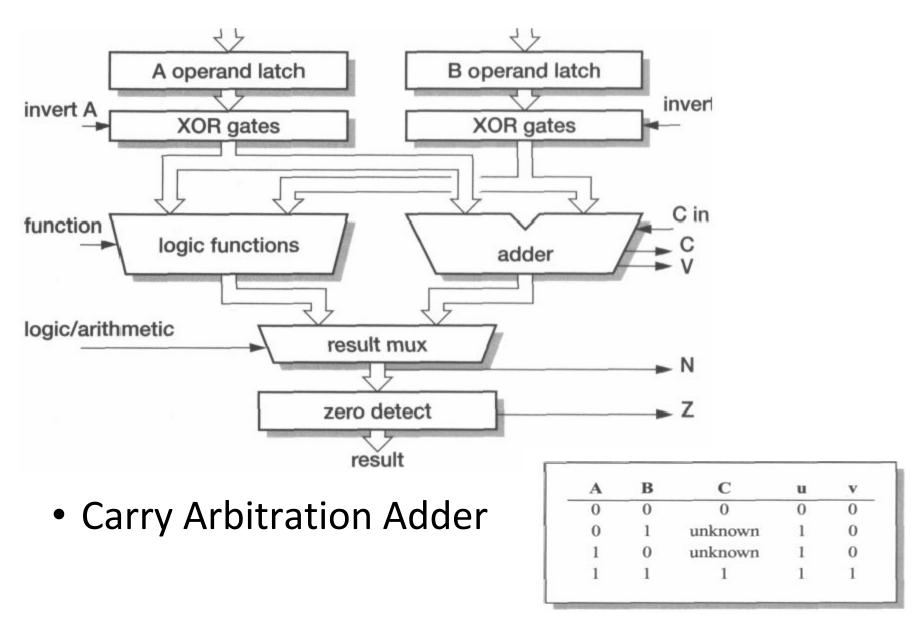
- Ripple Carry Adder
- Carry Look Ahead Adder
- Carry Select Adder
- Carry Arbitration Adder



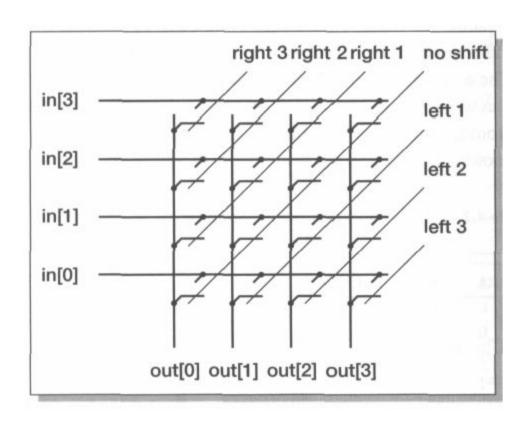
## **Carry Select Adder**



## **ALU Organization**

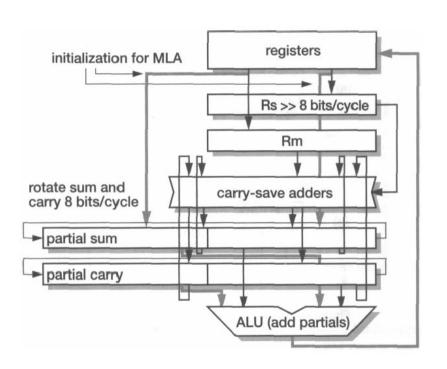


#### **Barrel Shifter**

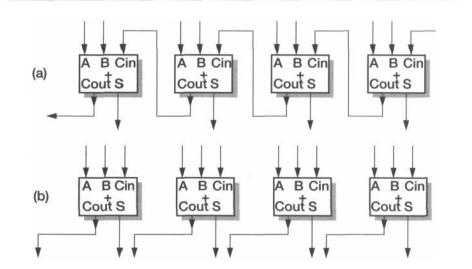


### Multiplier

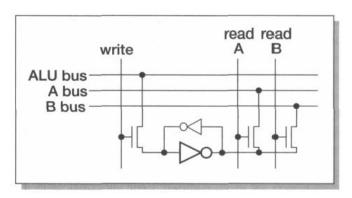
- Modified Booths Algorithm
- Carry Save adders



Carry-in	Multiplier	Shift	ALU	Carry-out
0	× 0	LSL #2N	A + 0	0
	$\times 1$	LSL #2N	A + B	0
	$\times 2$	LSL #(2N + 1)	A - B	1
	$\times 3$	LSL #2N	A - B	1
1	$\times 0$	LSL #2N	A + B	0
	$\times 1$	LSL #(2N + 1)	A + B	0
	×2	LSL #2N	A - B	1
	$\times 3$	LSL #2N	A + 0	1

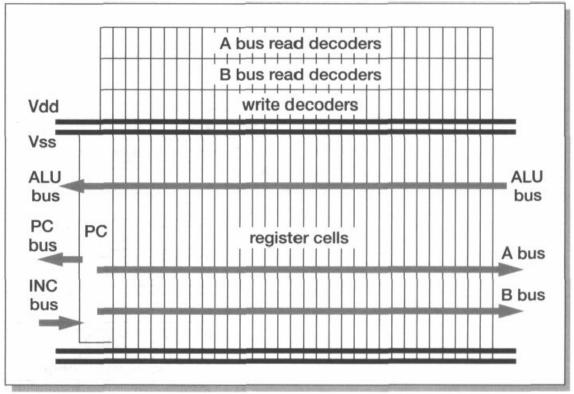


#### **ARM Register cell circuit and Layout**

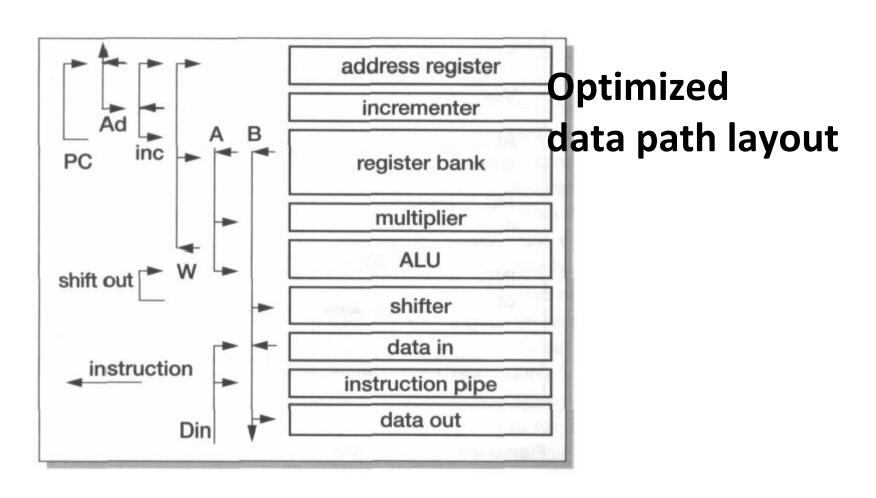


#### Register cell layout

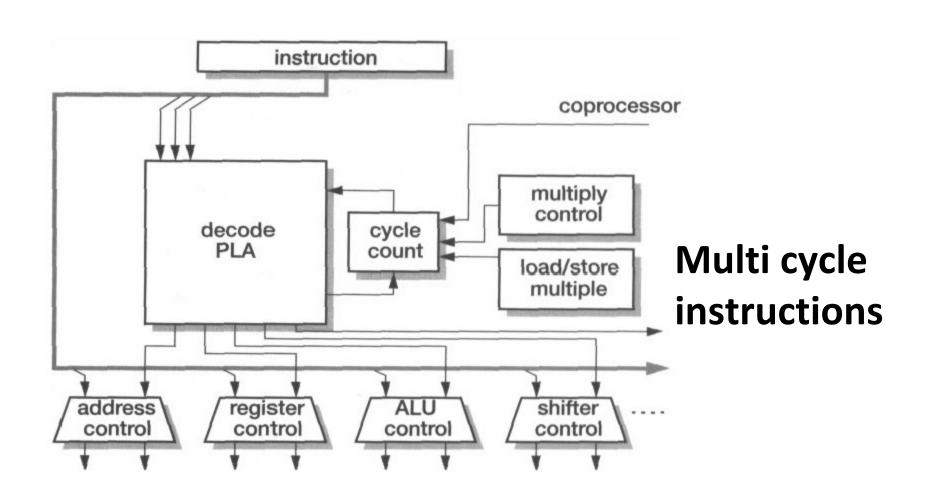
Register bank Ind ALU



### **Data Path Layout**



#### **ARM Control Structure**



#### Co Processor Interfacing

- ☐ 16 Logical coprocessors
- □ 16 registers not limited to 32 bit size
- Load store architecture

- □ CPB

#### Co Processor Instruction Execution

- ARM may decide not to execute it
- ARM decides to execute it, but no coprocessor available (CPA)
- ARM decides to execute it, coprocessor available, but busy (CPB)
- ARM decides to execute it, coprocessor accepts it for immediate execution