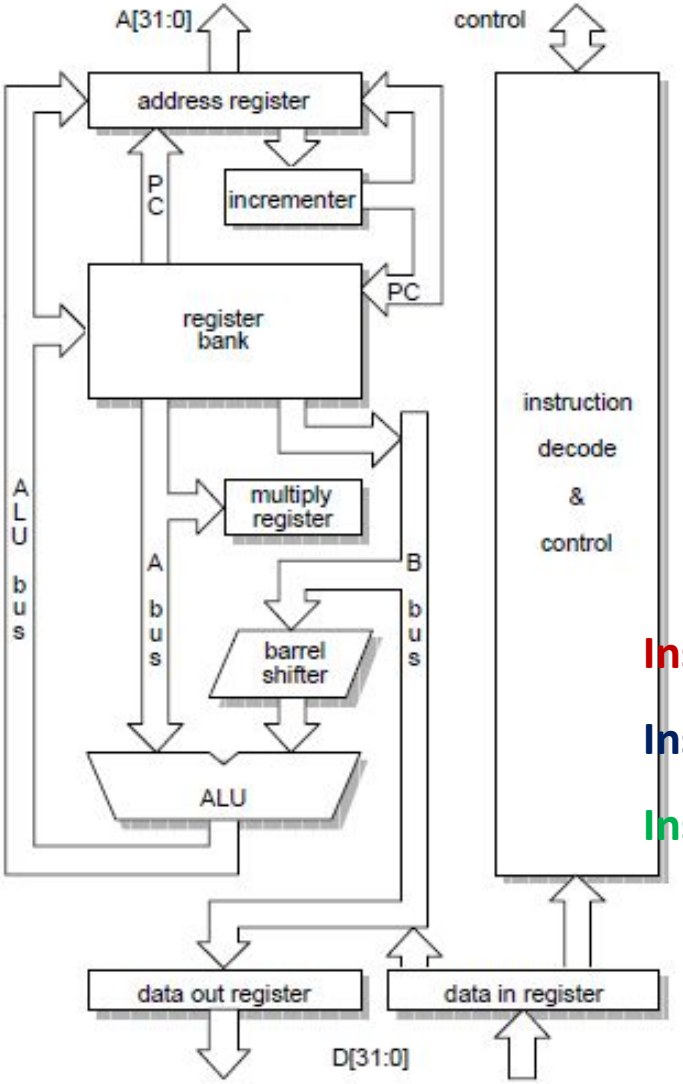
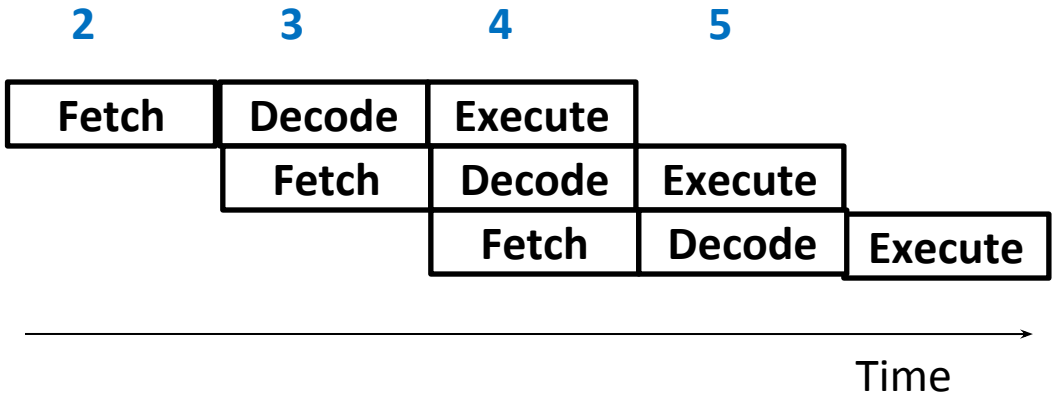
Embedded Systems

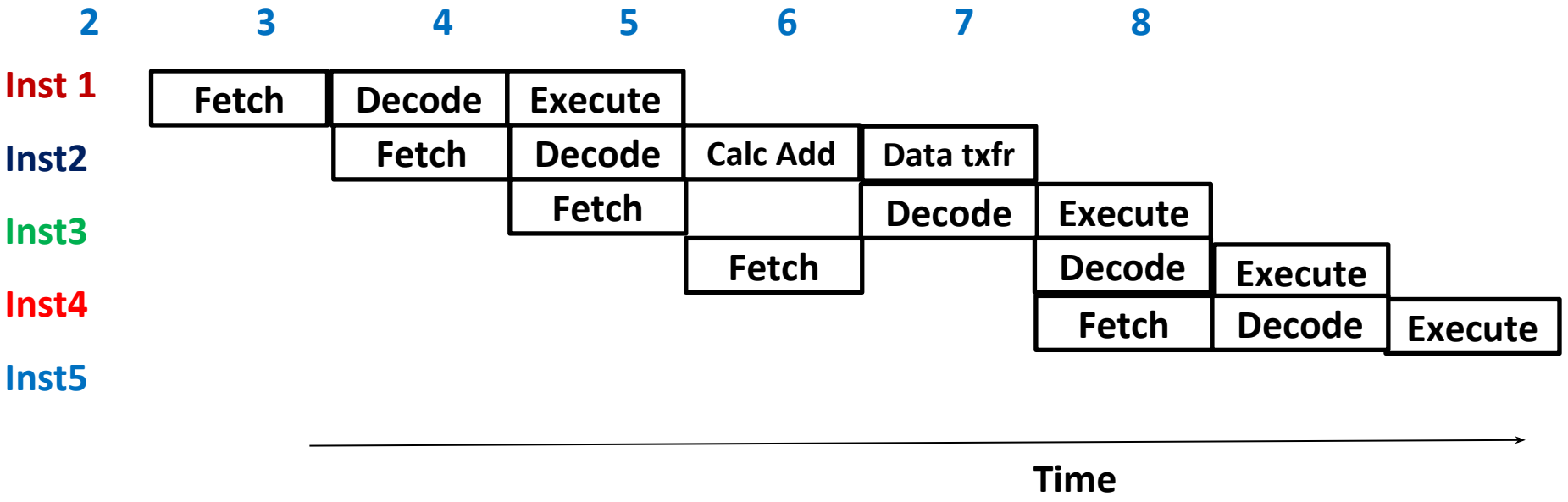
Dr. Sajesh Kumar U.

ARM 3 Stage Pipeline

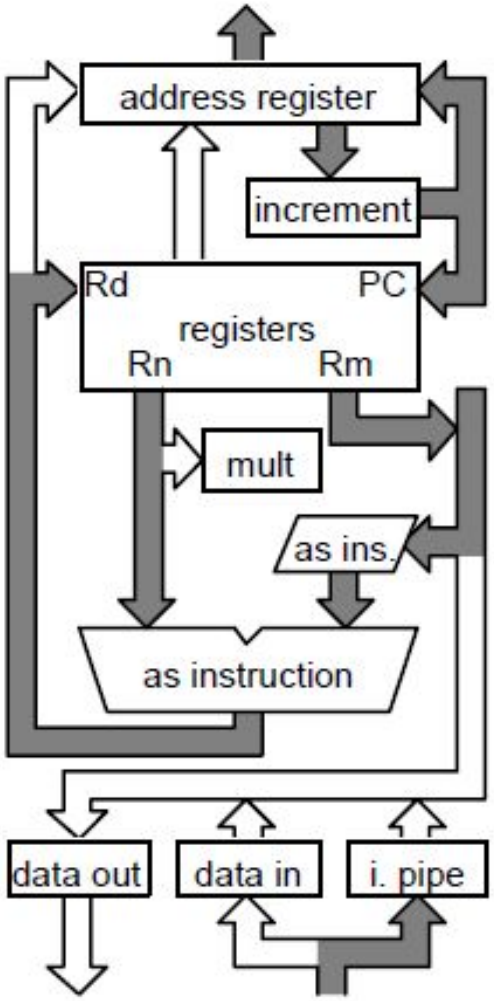




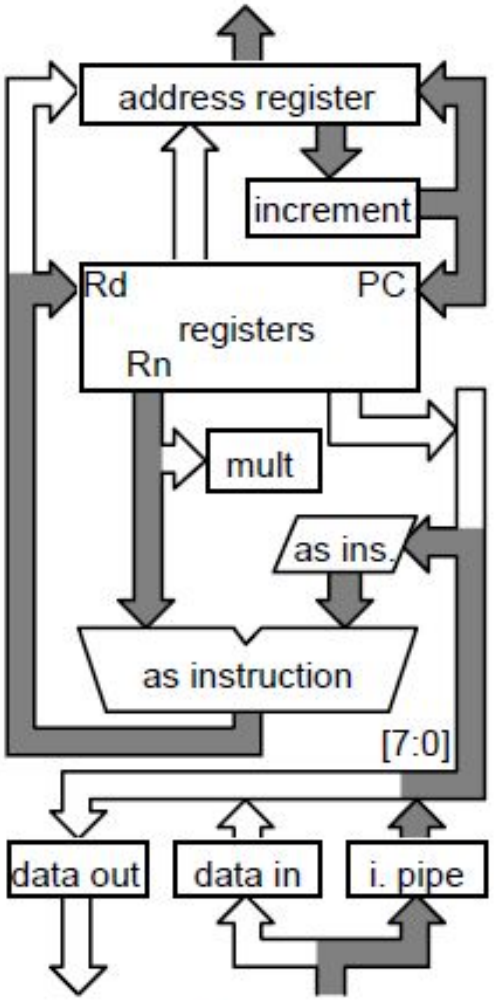
Multi cycle instruction in 3 stage pipeline



Data Processing Instruction-Data path Activity

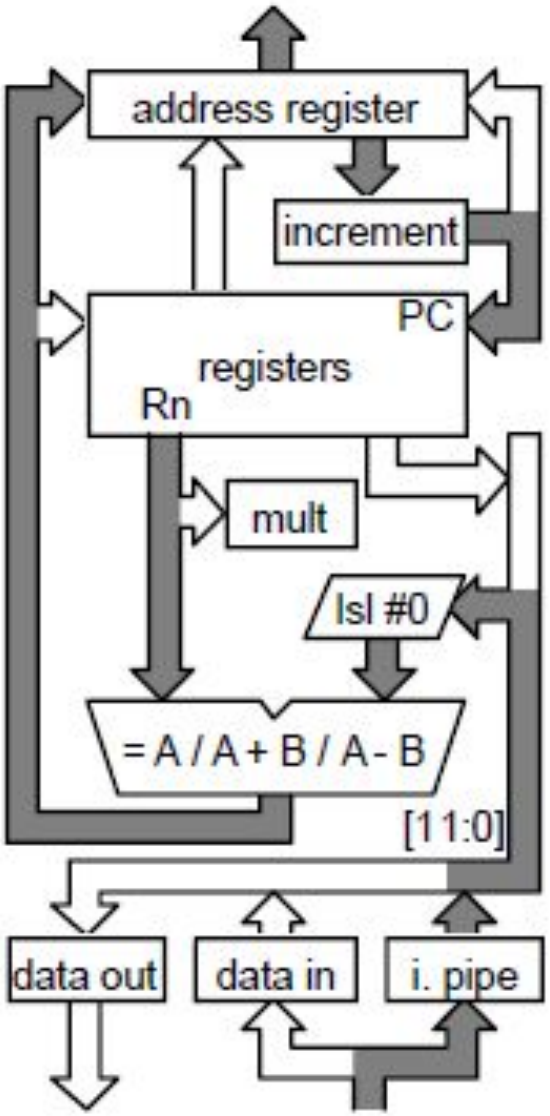


(a) register - register operations

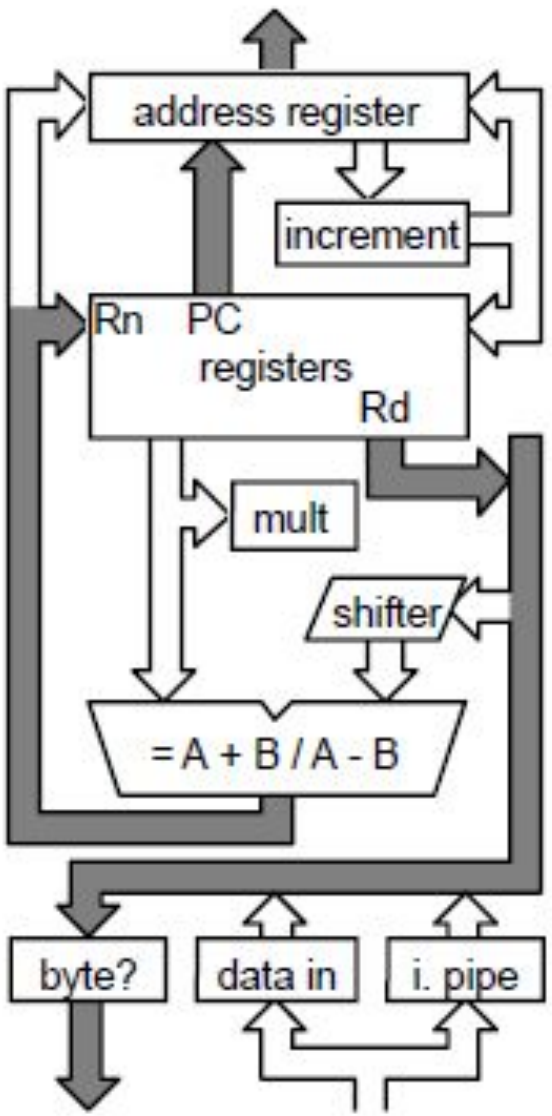


(b) register - immediate operations

Store Instruction- Data path activity

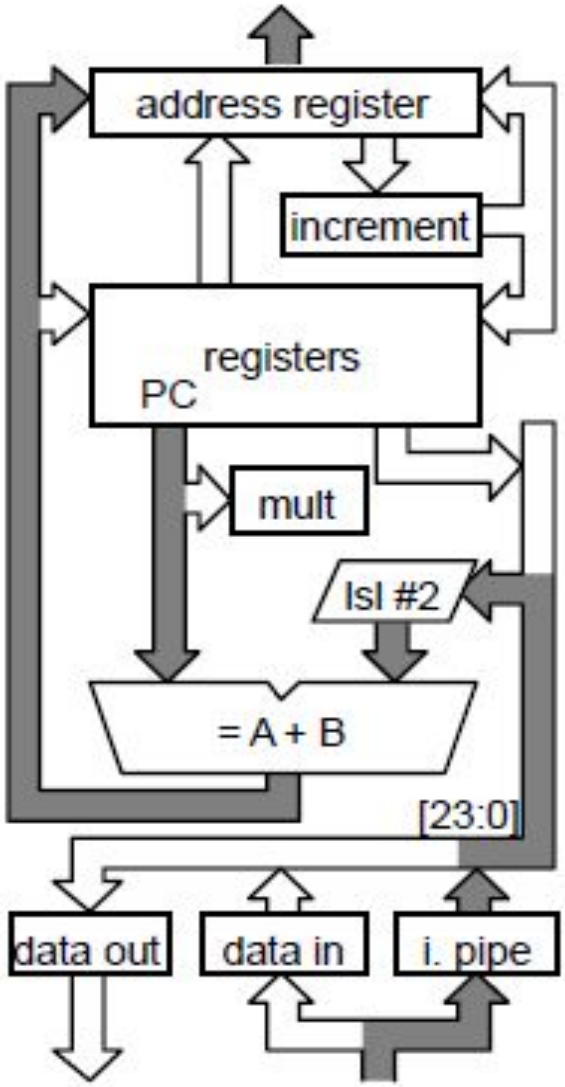


(a) 1st cycle - compute address

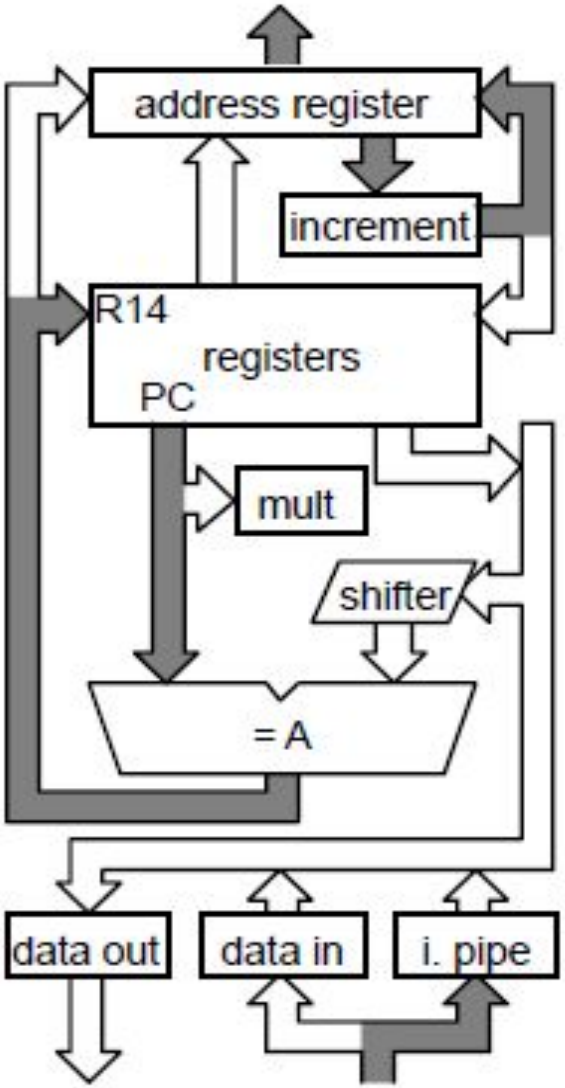


(b) 2nd cycle - store data & auto-index

Branch Instruction –Data path activity



(a) 1st cycle - compute branch target



(b) 2nd cycle - save return address

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

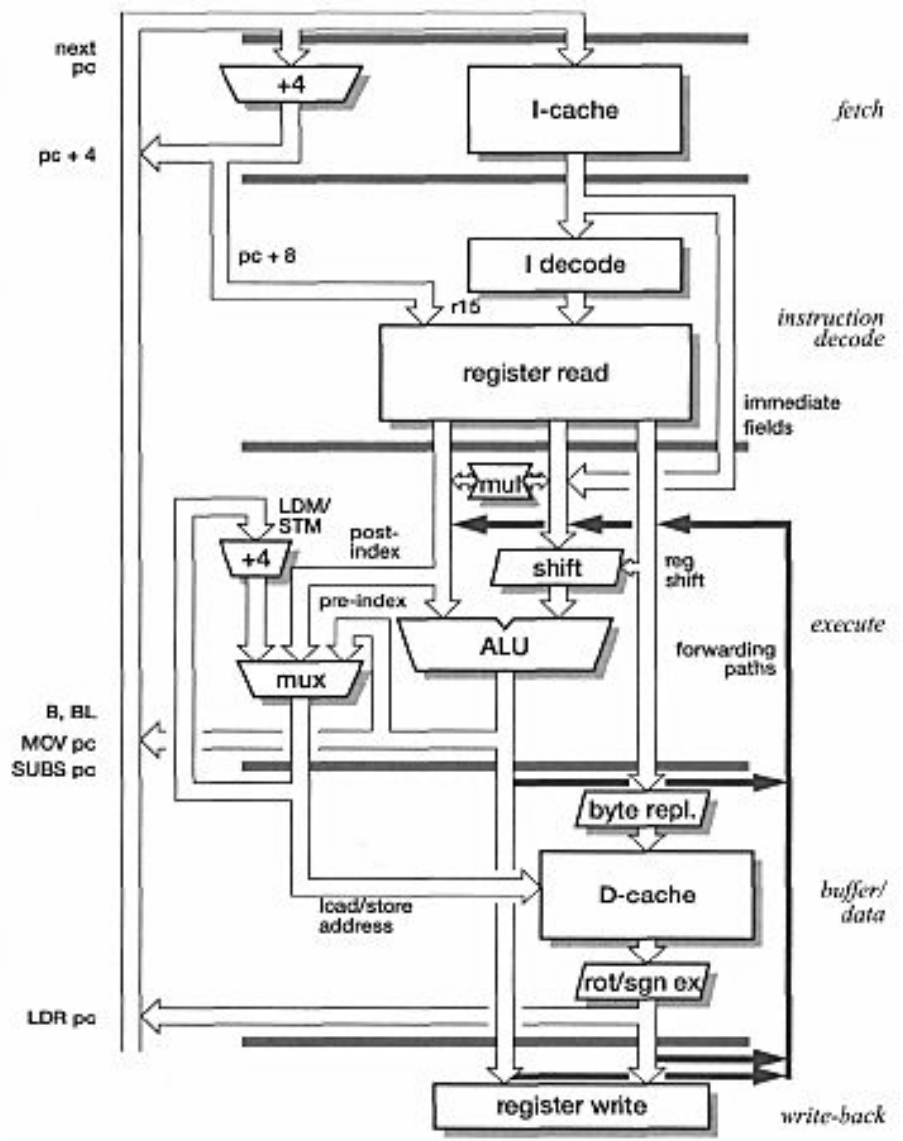


Figure 4.4 ARM9TDMI 5-stage pipeline organization.

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				stall	IF	ID	EX	MEM	WB

Data Hazards

Clock cycle number										
		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}	EX	MEM	WB			
AND	R6,R1,R7			IF	ID_{and}	EX	MEM	WB		
OR	R8,R1,R9				IF	ID_{or}	EX	MEM	WB	
XOR	R10,R1,R11					IF	ID_{xor}	EX	MEM	WB

Data forwarding mechanism

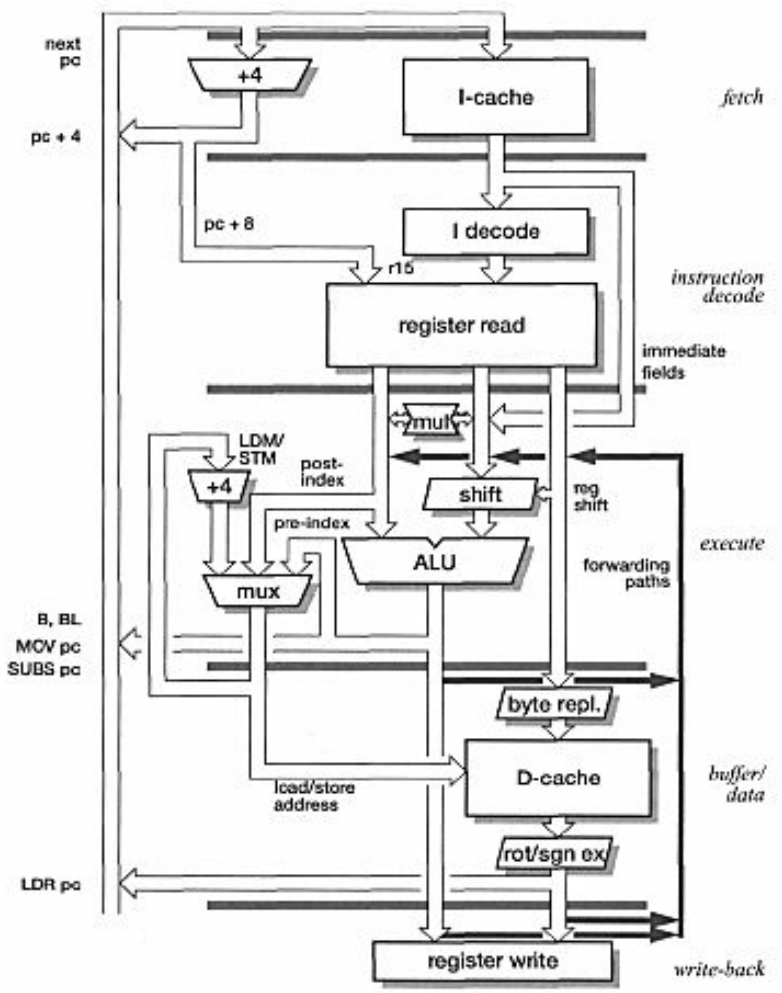


Figure 4.4 ARM9TDMI 5-stage pipeline organization.

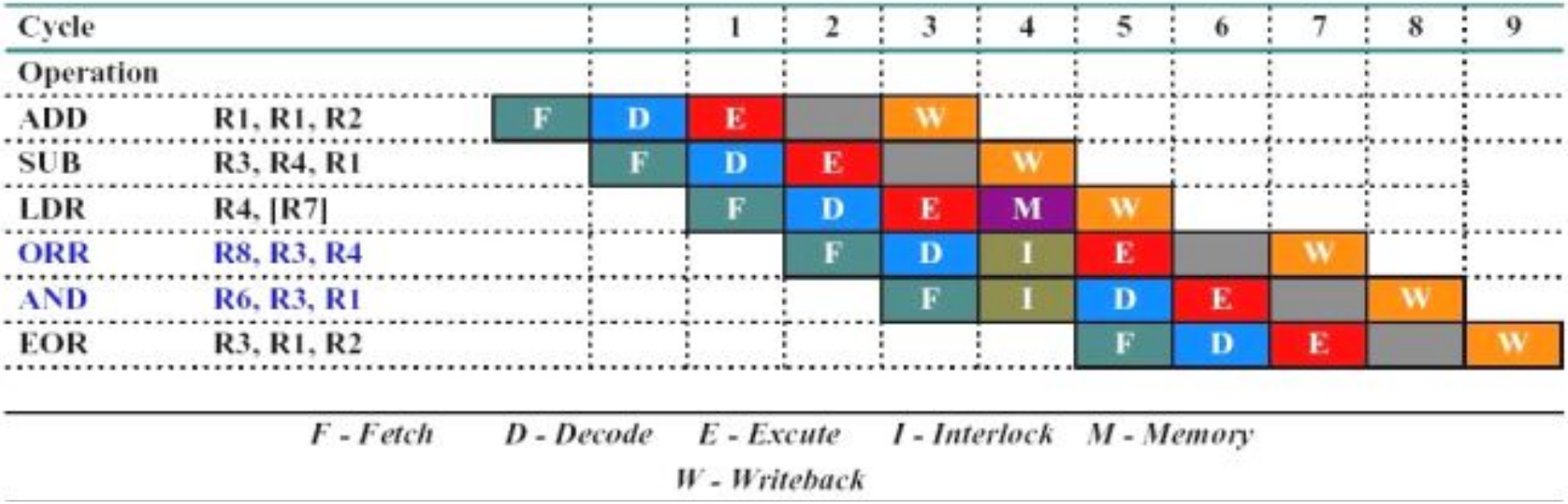
	Clock cycle number							
		1	2	3	4	5	6	7
ADD	R1,R2,R3	IF	ID	EX	MEM	**WB**		
SUB	R4,R5,R1		IF	ID_{sub}	EX	MEM	WB	
AND	R6,R1,R7			IF	ID_{and}	EX	MEM	WB

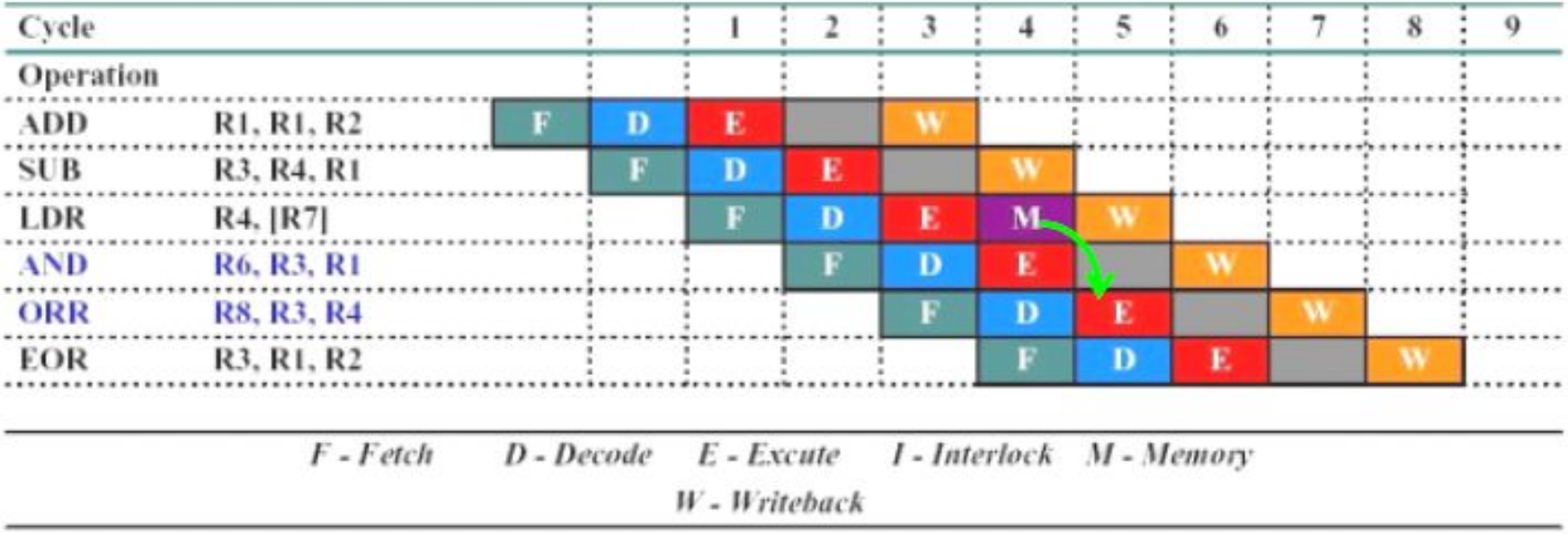
	Clock cycle number									
		1	2	3	4	5	6	7	8	9
ADD	R1,R2,R3	IF	ID	EX	MEM	**WB**				
SUB	R4,R5,R1		IF	*stall*	*stall*	ID_{sub}	EX	MEM	WB	
AND	R6,R1,R7			*stall*	*stall*	IF	ID_{and}	EX	MEM	WB

Memory Access and instruction Stall

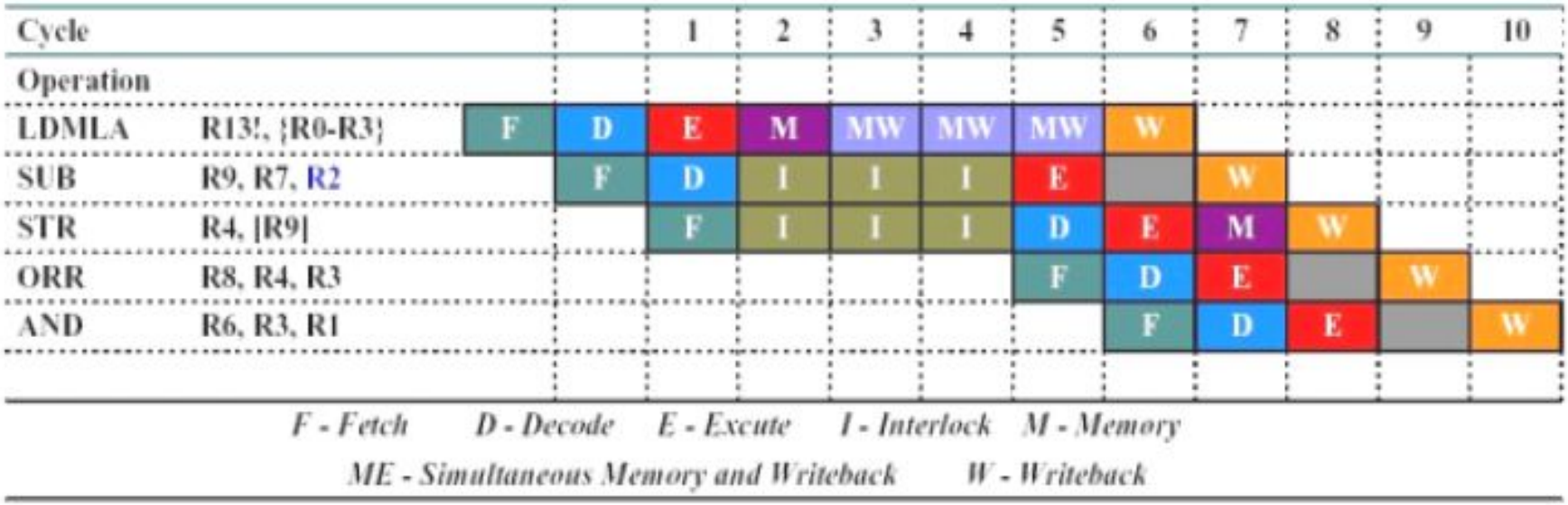
		1	2	3	4	5	6	7	8	9
LDR	R1,@(R2)	IF	ID	EX	MEM	WB				
SUB	R4,R1,R5		IF	ID	*stall*	EX_{sub}	MEM	WB		
AND	R6,R1,R7			IF	*stall*	ID	EX	MEM	WB	
OR	R8,R1,R9				*stall*	IF	ID	EX	MEM	WB

Load Interlock

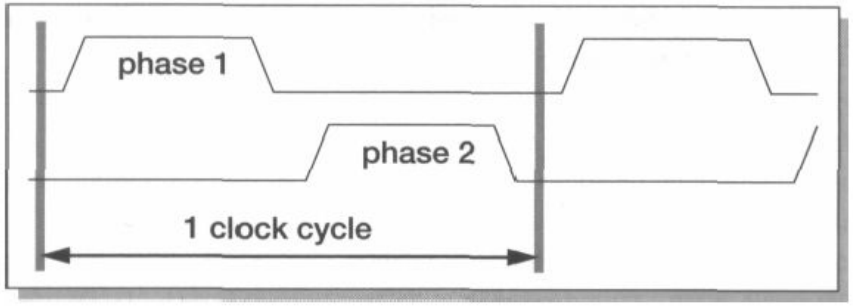




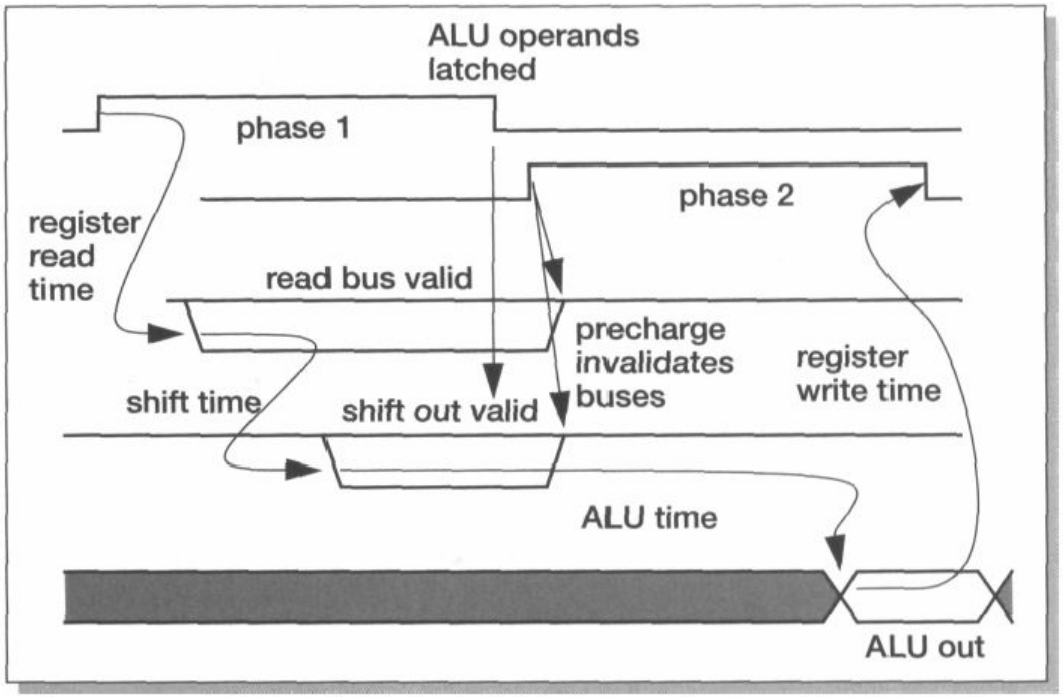
LDM Interlock



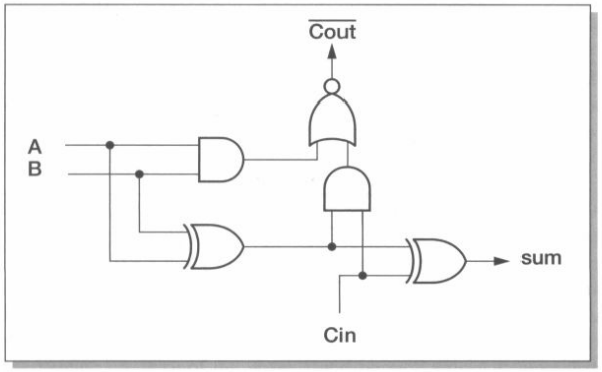
ARM Implementation

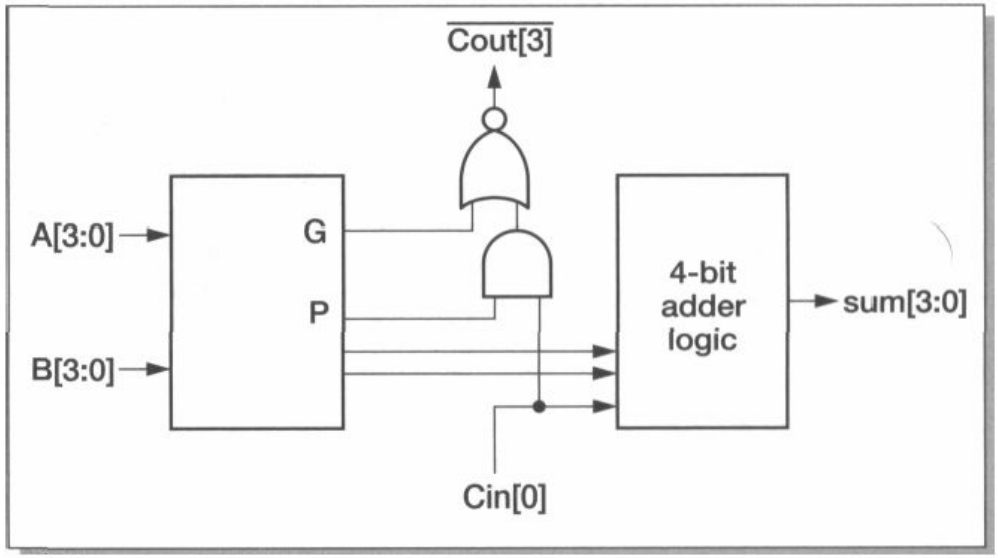


- **Register read buses are dynamic**
- **Two phase non overlapping clock**

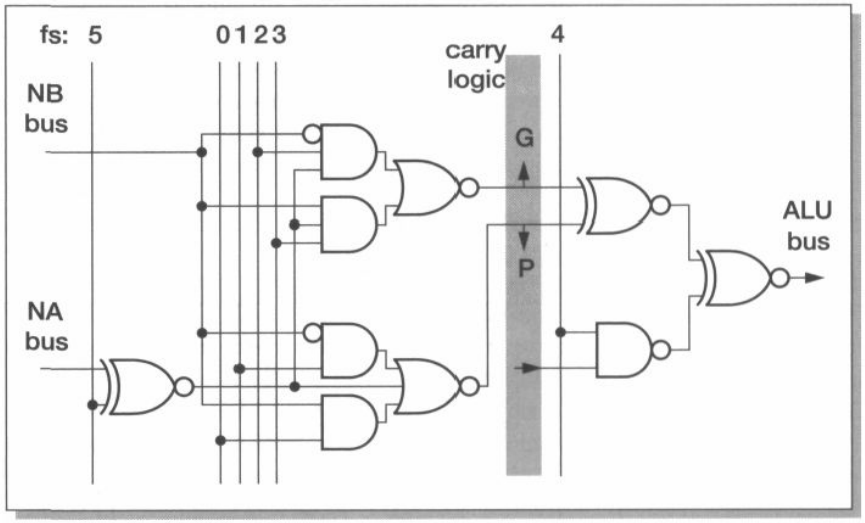


Adder Design

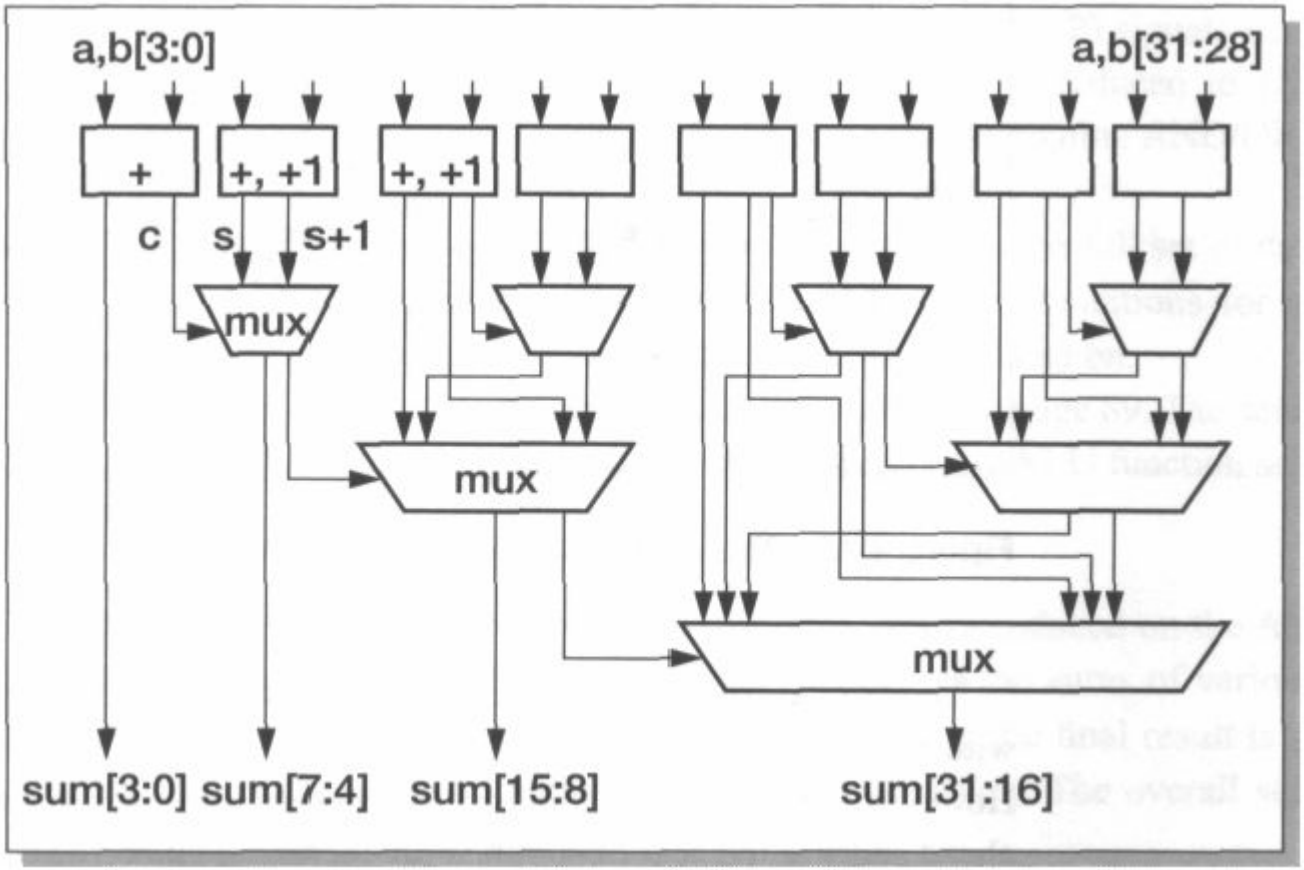




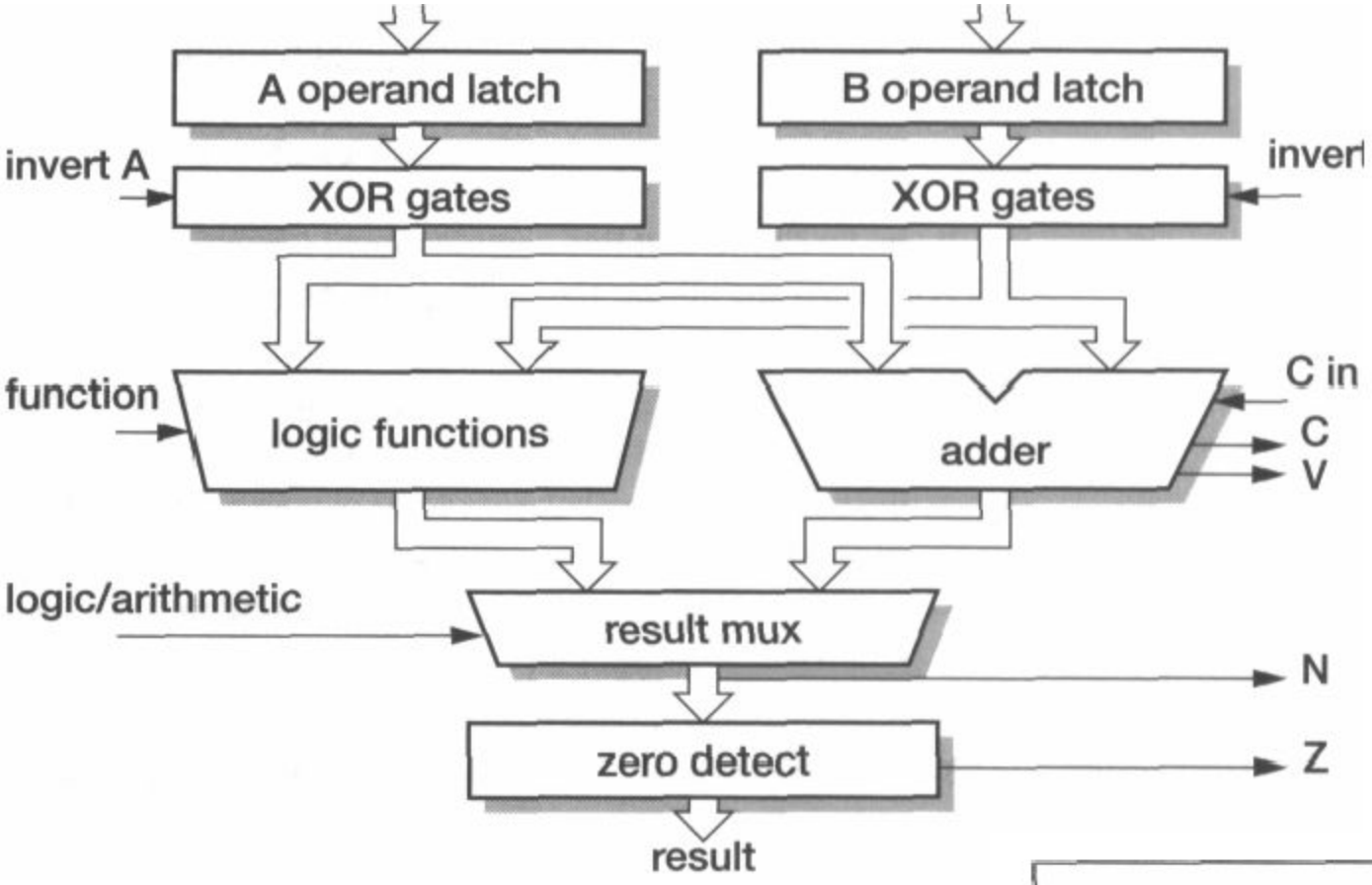
- Ripple Carry Adder
- Carry Look Ahead Adder
- Carry Select Adder
- Carry Arbitration Adder



Carry Select Adder



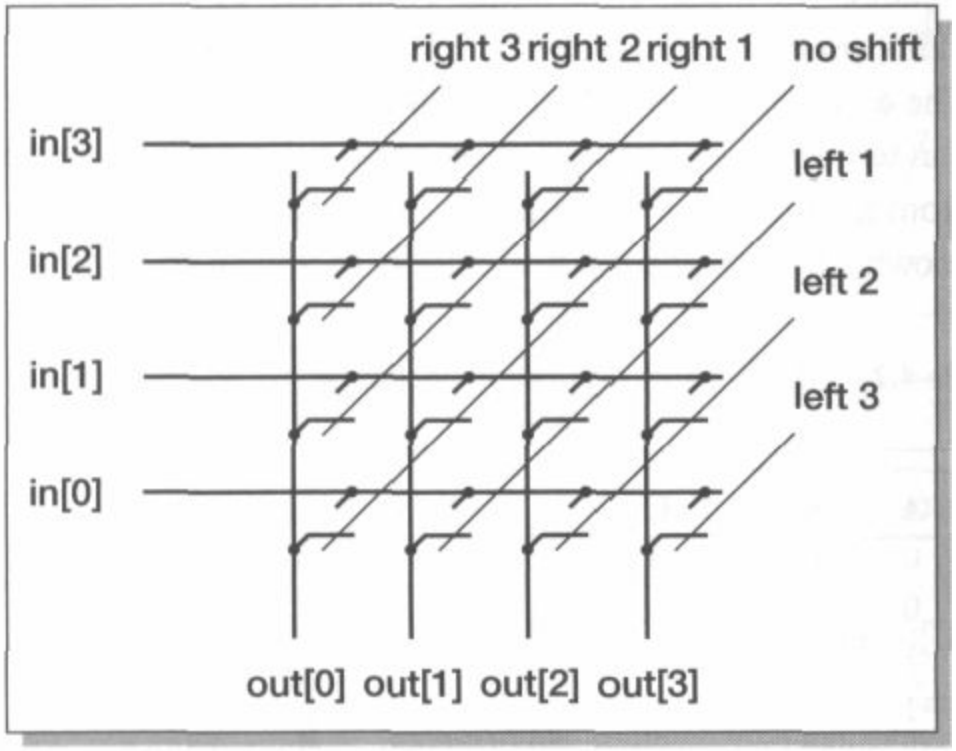
ALU Organization



- Carry Arbitration Adder

A	B	C	u	v
0	0	0	0	0
0	1	unknown	1	0
1	0	unknown	1	0
1	1	1	1	1

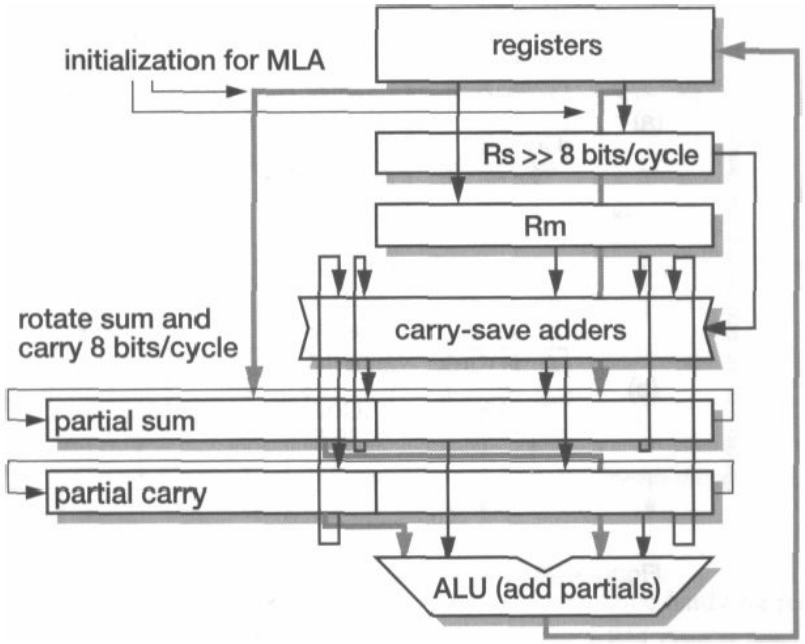
Barrel Shifter

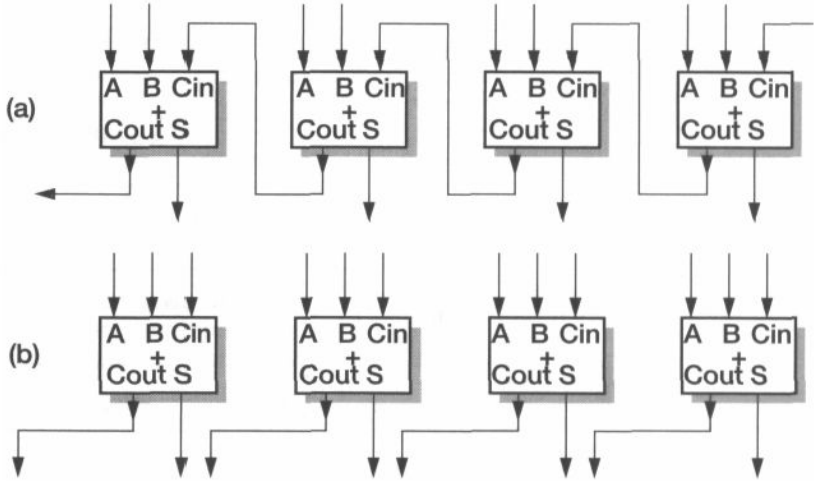


Multiplier

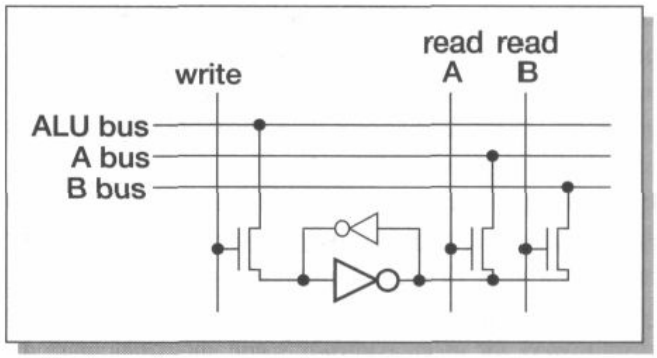
- Modified Booths Algorithm
- Carry Save adders

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

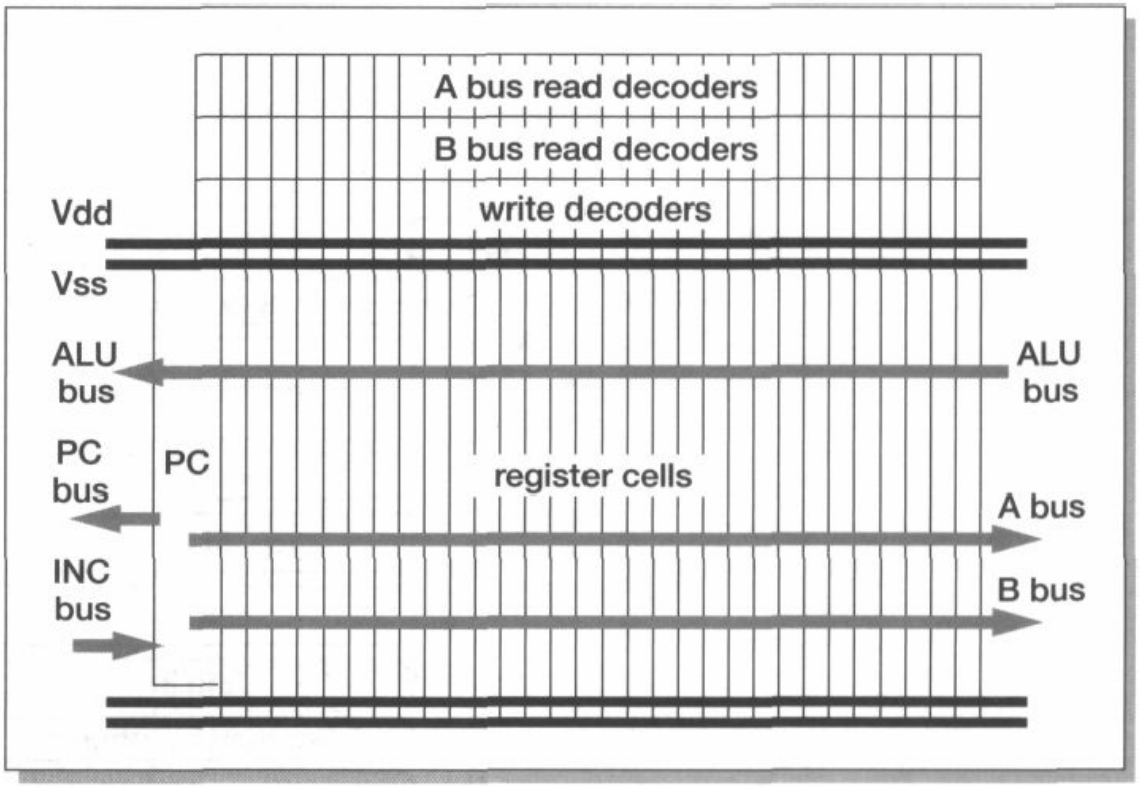




ARM Register cell circuit and Layout



Register cell layout

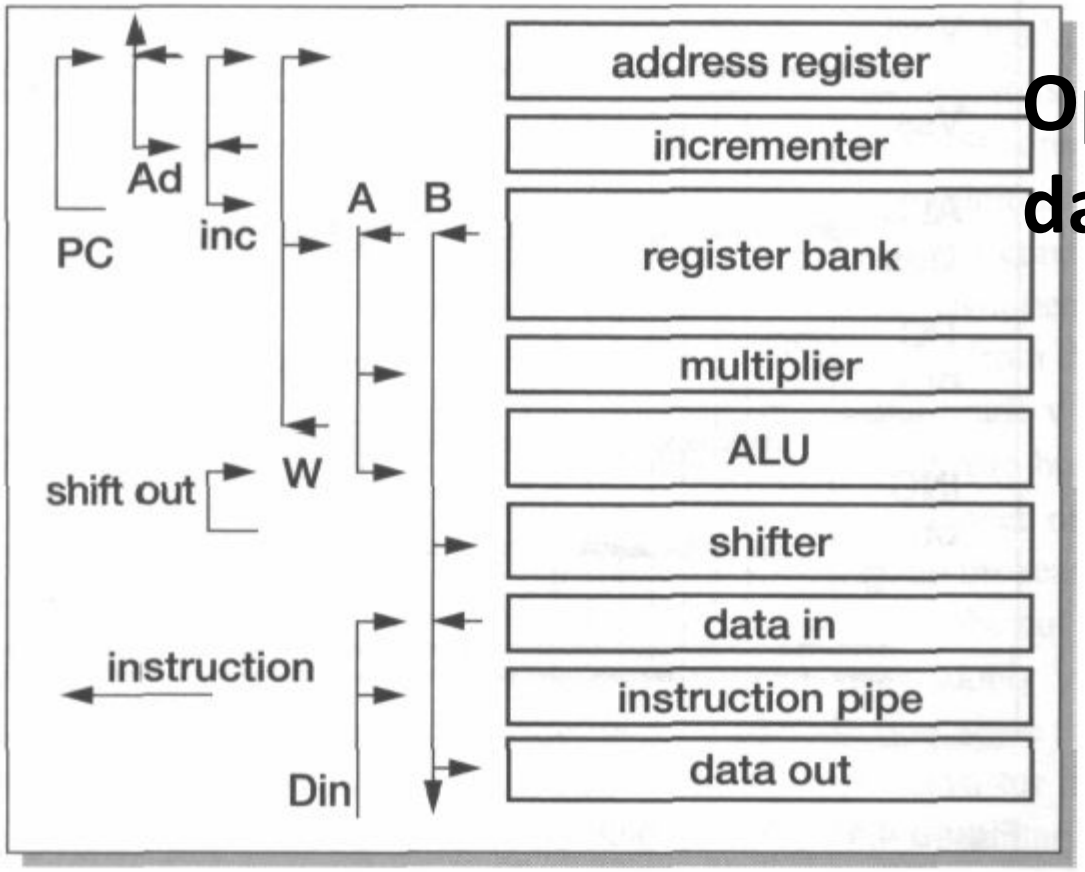


Register bank

and

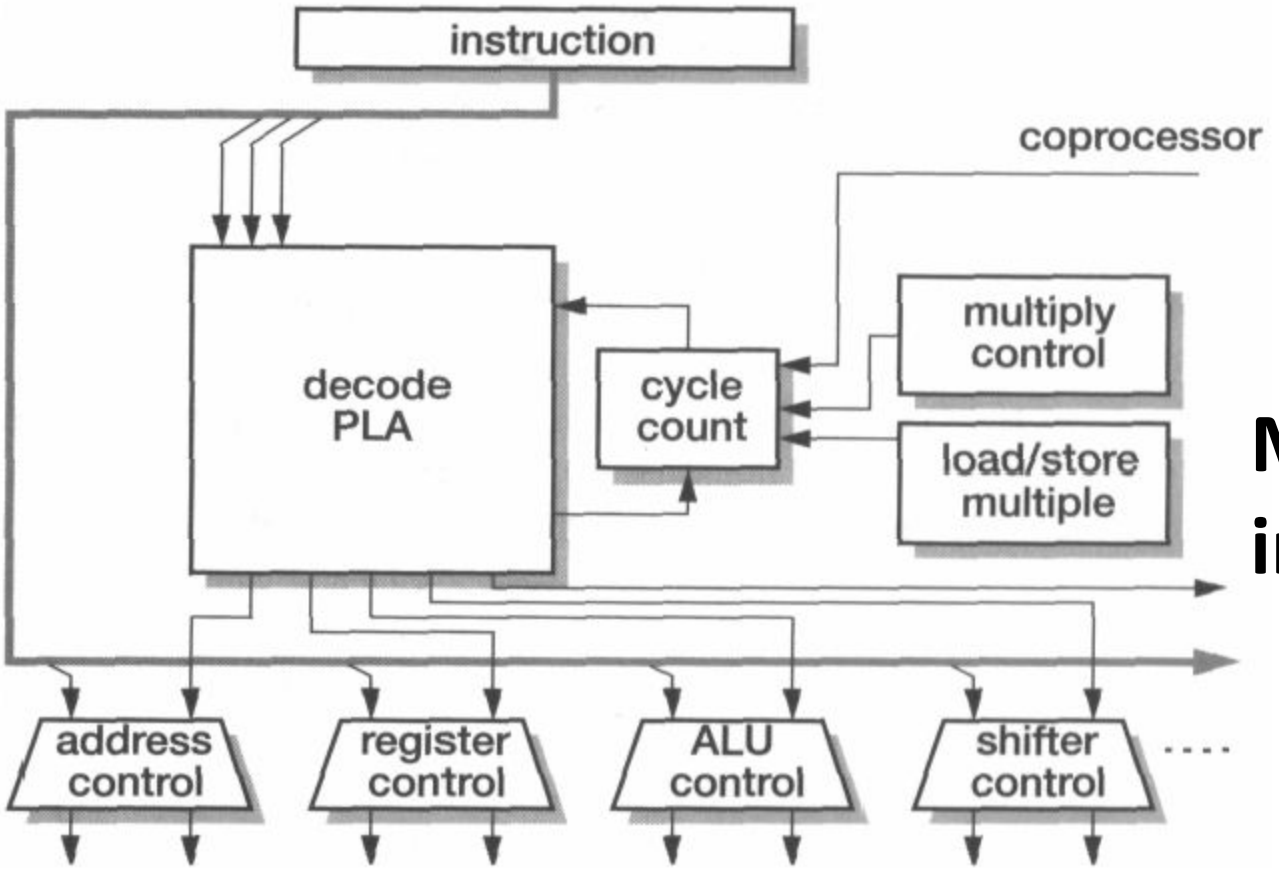
ALU

Data Path Layout



Optimized data path layout

ARM Control Structure



Multi cycle instructions

Co Processor Interfacing

- 16 Logical coprocessors
- 16 registers not limited to 32 bit size
- Load store architecture
- CPI
- CPA
- 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