

# Microprogram Design

- ▶ Use ASM from Lecture 16 to design the Microprograms
  - ▶ See Symbolic/Binary Microprogram on next slide
- <https://powcoder.com>
- Add WeChat powcoder

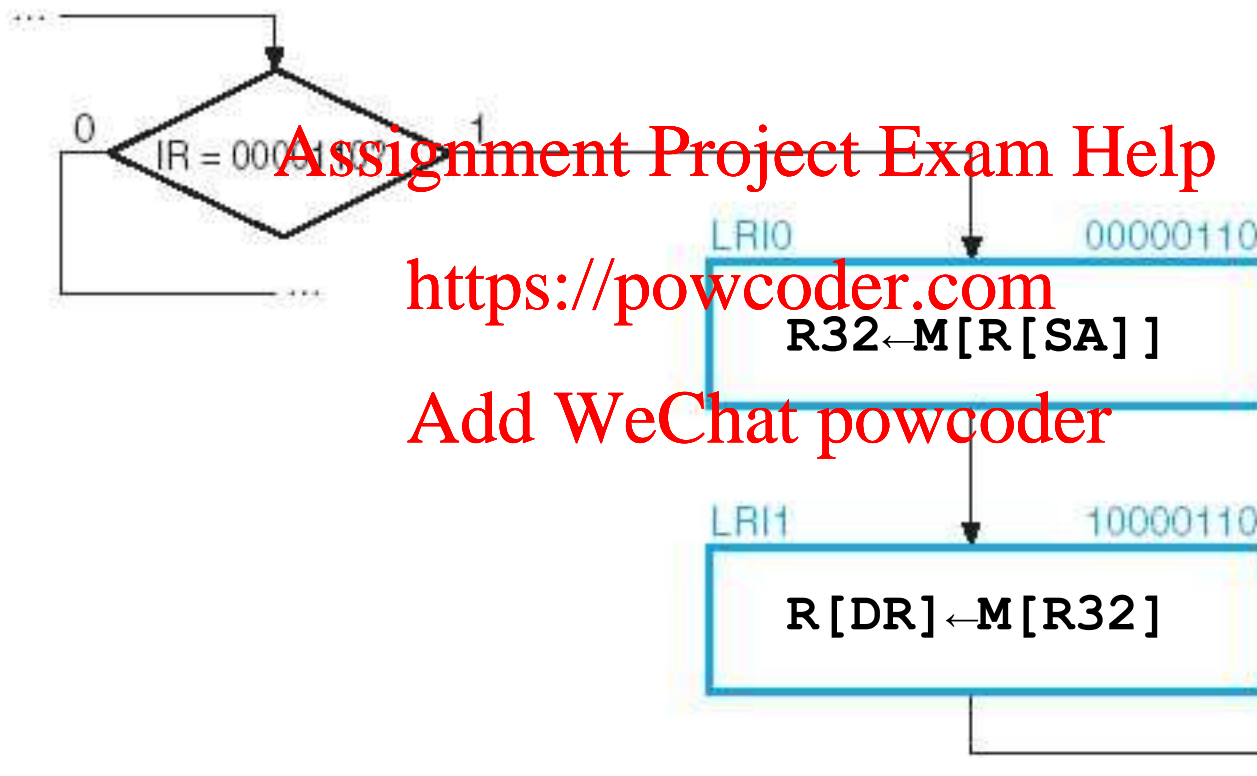
# Symbolic/Binary Microprogram

Address	NXT ADD	MS	MC	IL	PI	PL	TD	TA	TB	MB	FS	MD	RW	MM	MW
IF	EX0	CNT	—	LDI	INP	NLP	—	—	—	—	—	—	NW	PC	NW
EXO	—	NXT	OPC	NLI	NLP	NLP	—	—	—	—	—	—	NW	—	NW
ADI	IF	NXT	NXA	NLI	NLP	NLP	DR	SA	—	—	Constant $A+B$	FnUt	WR	—	NW
LD	IF	NXT	NXA	NLI	NLP	NLP	DR	SA	—	—	—	Data	WR	MA	NW
ST	IF	NXT	NXA	NLI	NLP	NLP	—	SA	SB	Register	—	—	NW	MA	WR
INC	IF	NXT	NXA	NLI	NLP	NLP	DR	SA	—	—	$F = A + 1$	FnUt	WR	—	NW
NOT	IF	NXT	NXA	NLI	NLP	NLP	DR	SA	—	—	$F = \bar{A}$	FnUt	WR	—	NW
ADD	IF	NXT	NXA	NLI	NLP	NLP	DR	SA	SB	Register	$F = A + B$	FnUt	WR	—	NW

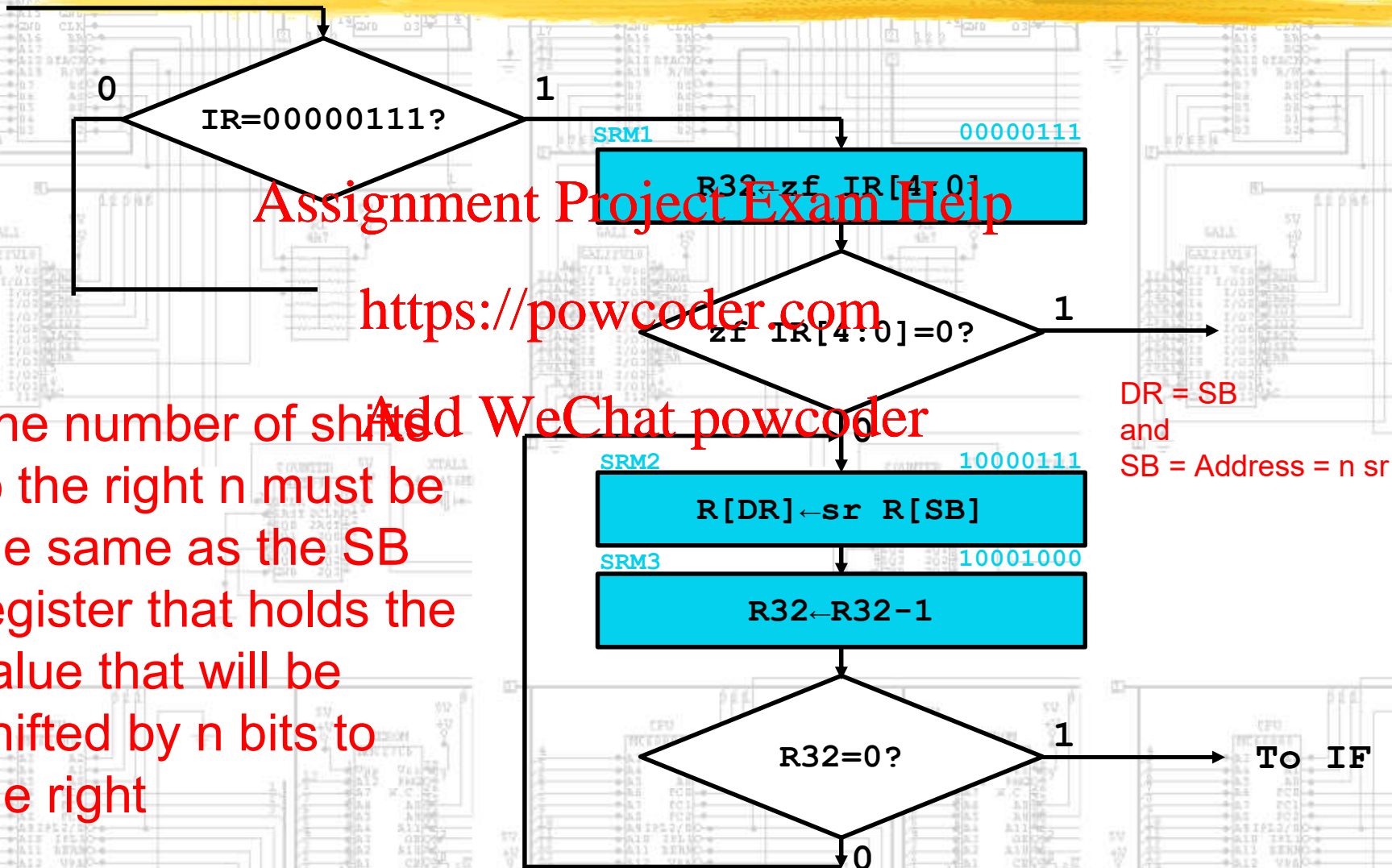
Address	NXT ADD	MS	MC	IL	PI	PL	TD	TA	TB	MB	FS	MD	RW	MM	MW
192	193	000	0	1	1	0	0	0	0	0	00000	0	0	1	0
193	000	001	1	0	0	0	0	0	0	0	00000	0	0	0	0
000	192	001	0	0	0	0	0	0	0	1	00010	0	1	0	0
001	192	001	0	0	0	0	0	0	0	0	00000	1	1	0	0
002	192	001	0	0	0	0	0	0	0	0	00000	0	0	0	1
003	192	001	0	0	0	0	0	0	0	0	00001	0	1	0	0
004	192	001	0	0	0	0	0	0	0	0	01110	0	1	0	0
005	192	001	0	0	0	0	0	0	0	0	00010	0	1	0	0

This is not our control memory but similar

# Indirect Instruction ASM

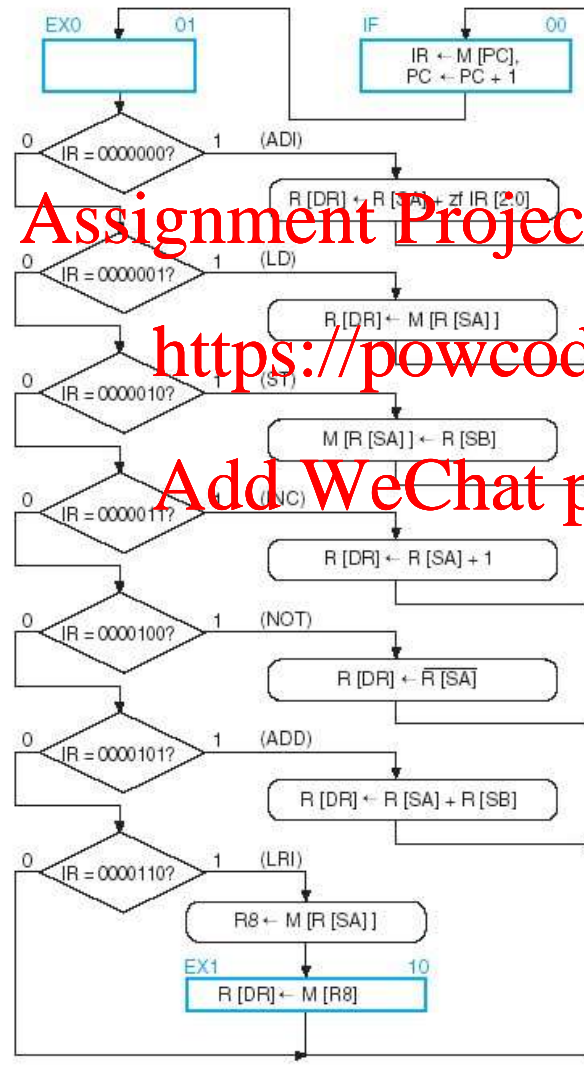


# Right-Shift Instruction ASM





# Hardwired Multiple-Cycle Control

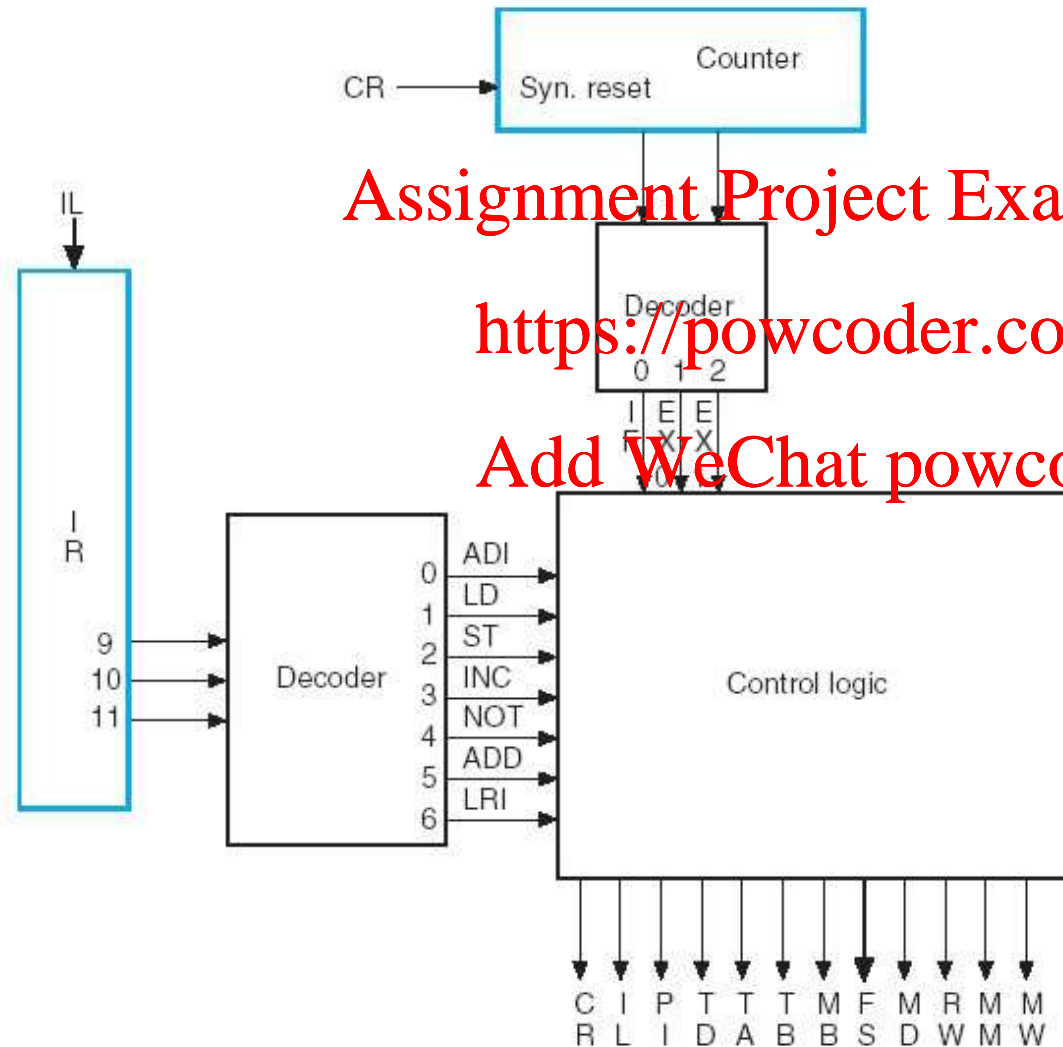


Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

# Hardwired Control Unit



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

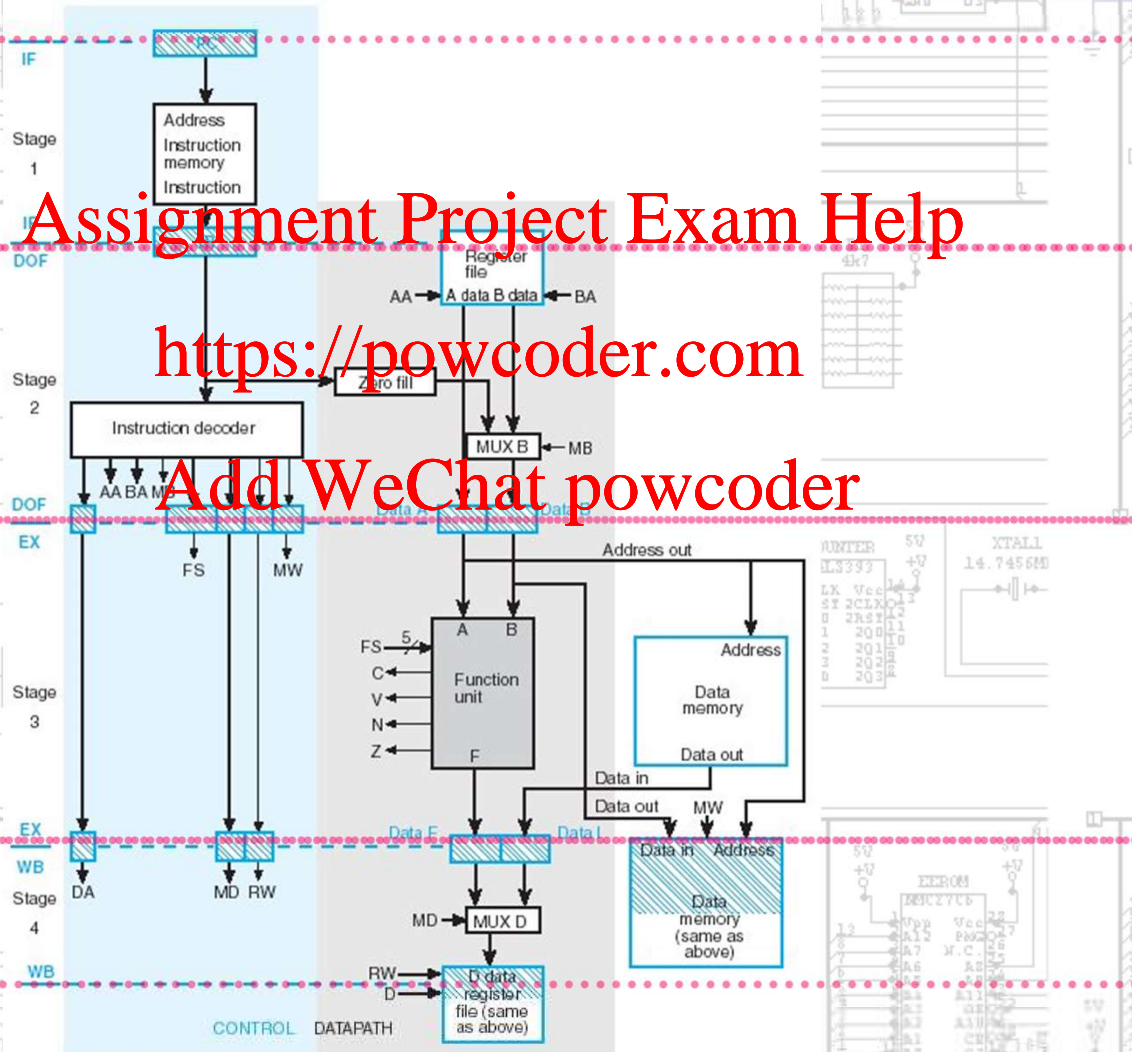
# Pipelined (based on single-cycle)

Instruction Fetch

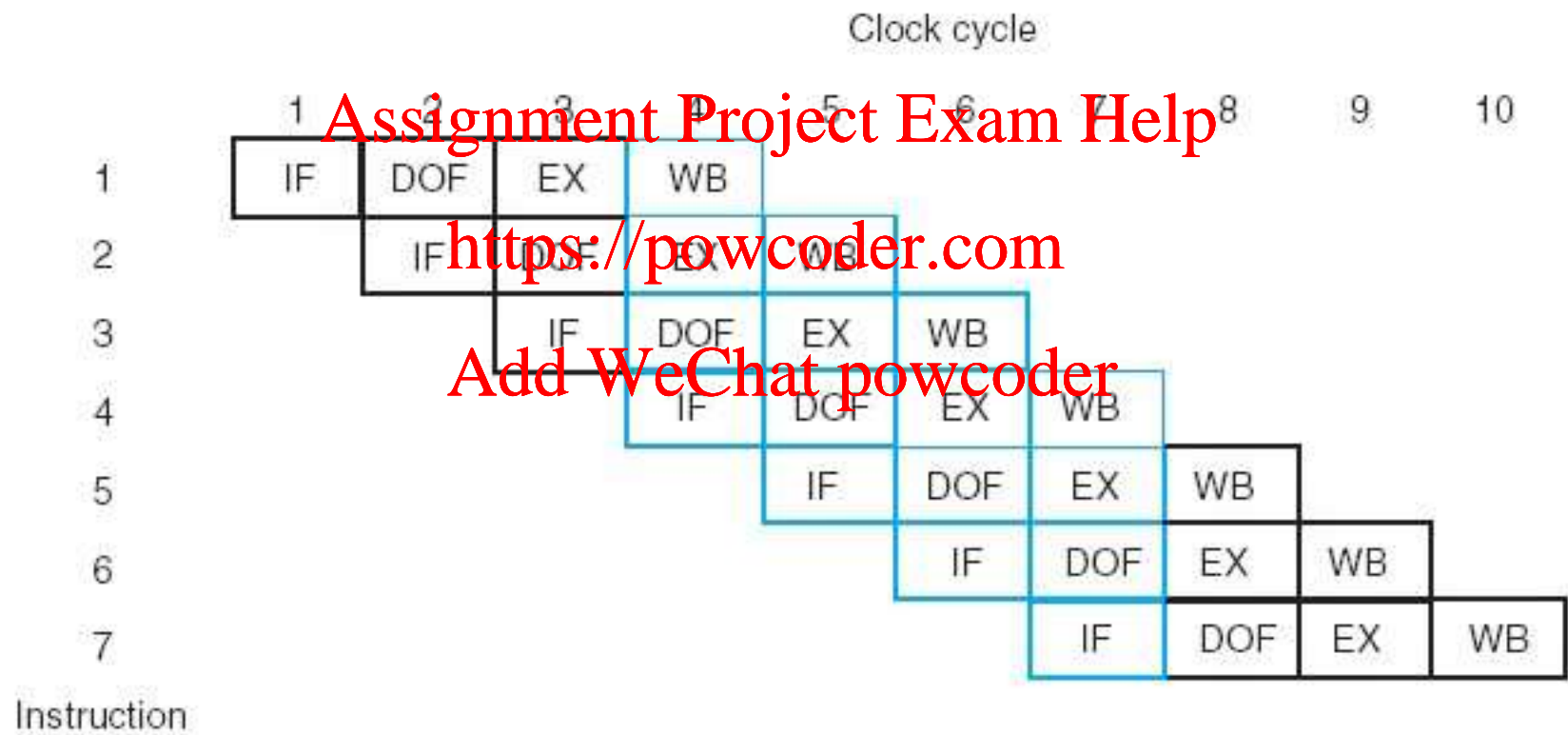
Decode and  
Operand Fetch

Execution

Write-back



# Pipelined Execution Pattern





# Computer Architecture and Microprocessor Systems

