Mic-1 Microprogram

-fetch decode execute

Sequence of Micro instructions that interpret IJVM ISA instruction (Java bytecode)

e.g. Java: n = 25 + 15;

		IJVM Binary	Byte stream	
IJVM:	~~BIPUSH 0x19~~	0x10 0x19	PC → 0x10	1 ← PC (PC = 0)
	BIPUSH 0x0F	0x10 0x0F	0x19	2
	IADD	0x60	0x10	3
	ISTORE 0x01	0x36 0x01	0x0F	4
			0x60	5
			0x36	6
			0x01	7

hex	int
0x10 ⇒	16_{10}
0x60 ⇒	96_{10}
0x36 ⇒	54_{10}

To interpret BIPUSH IJVM instruction, several micro instructions:

bipush1:

bipush2:

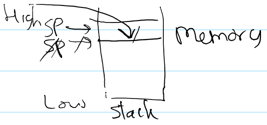
bipush3:

SP = SP + 1

~~MAR~~ =

MDR

Read/write



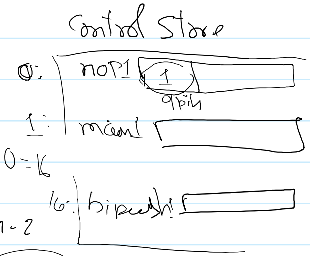
? How microprogram begins the execution

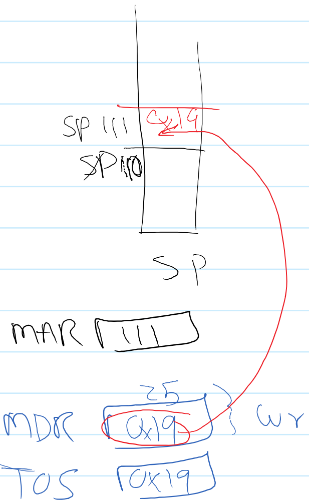
At boot time all registers will hold zeroes. PC, MBR, MPC

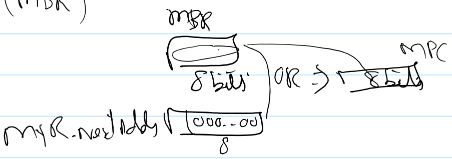
	CPU cycle		MPC	PC	MBR	
boot	✓1	nop1	0	0	0	
	✓2	main1	~~1~~ 0	~~0~~ 1	0	Start fetch → finish } 0x10 = 16
	✓3	nop1	0			
1st IJVM interpret	4	main1	1	~~1~~ 2	0x10 = 16	start fetch 2
	5	bipush1:	16 (0x10 16)			MAR → 0x19 = 25 opcode
	6	bipush2	17	~~2~~ 3	0x19	start → fetch3
	7	bipush3	18		0x19	
2nd IJVM interpretation	8					
	9					
	10					
	11					

micro instruction: ~~0x10~~ 16

goto (MBR)







IADD

12 main1

13 iadd1

14 iadd2

15 iadd3

ISTORE

16 main1

17 istore1

18 istore2

. . .

22 istore6

To interpret an IJVM instruction
The microprogram executes

main1
opcode1
opcode2
⋮
opcode6 } microinstructions.

microinstruction

e.g IADD ⇒ main1
iadd1
iadd2
iadd3

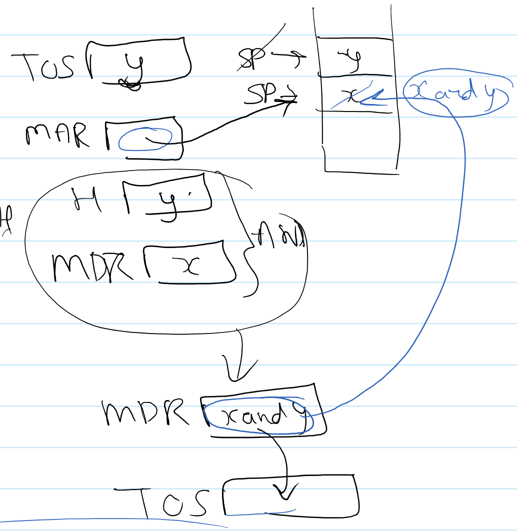
INVOKEVIRTUAL

main1
invokevirtual1
invokevirtual2
⋮
invokevirtual22 } 23 microinstructions

IAND

iand1 MAR=SP=SP-1; rd
MDR
iand2 H=TOS
iand3 MDR=TOS=MDR and H
wr;

main1:


TOS y
SP
y
SP
x
x and y
MAR
H y
MDR x
AND
MDR x and y
TOS