**CDA 4101** **Worksheet: IJVM Microcode** **Spring 2017**

Write MAL code fragments to implement the IJVM instructions. There are usually more than 1 implementation. Expect to make errors!

IJVM Instruction Set

* Idle the Processor

x00 **NOP** ;do nothing for 1 cycle

* Data Movement

x10 **BIPUSH** *byte* ;push immediate byte

x13 **LDC\_W** *index* ;push word from the constant pool

x15 **ILOAD** *varnum* ;push word from a local variable

x36 **ISTORE** *varnum* ;pop word to a local variable

* Stack Manipulation

x57 **POP** ;remove and discard the top word

x59 **DUP** ;duplicate the top word

x5F **SWAP** ;exchange the top two words

* Data Operations

x60 **IADD** ;pop two words, add, push result

x64 **ISUB** ;pop two words, subtract, push result

x7E **IAND** ;pop two words, bitwise and, push result

x80 **IOR** ;pop two words, bitwise or, push result

x84 **IINC** *varnum* *const*

* Branching

x99 **IFEQ** *offset* ;conditional, if top word = 0

x9B **IFLT** *offset* ;conditional, if top word < 0

x9F **IF\_ICMPEQ** *offset* ;conditional, if two top words equal

xA7 **GOT**O *offset* ;unconditional

* Call & Return

xAC **IRETURN** ;push return value, return from method call

xB6 **INVOKEVIRTUAL** *disp* ;invoke a subroutine

* Prefix Instruction

xC4 **WIDE** ;extend index of load or store

Notes

1. Easiest first order: **NOP**, Stack Manipulation, Data Operations, Data Movement, **WIDE**, Branching, Call & Return
2. Operands. 1 byte: *byte, const, varnum*. 2 bytes: *disp, index, offset*.
3. At the start of any instruction, the opcode has been read and the byte following the opcode is available in MBR.
4. The last microinstruction of any instruction specifies Main1 as the next address.
5. All stack operations leave a copy of the top word in **TOS**.
6. **Memory access takes a complete cycle!!!**
7. Reconcile your solutions with the microprogram given in **Figure 4-17**, pages 272-4.