

Department of Electrical and Computer Engineering
The University of Texas at Austin

Yale Patt, Instructor
Hyesoon Kim, Moinuddin Qureshi, TAs

Project Specification Summary and Clarifications

Instructions

=====

ADD (p. 3-21) -- All variations
OR (p. 3-511) -- All variatons
NOT (p. 3-509) -- All variations
MOV (p. 3-432) -- All variations except moffs* format
DEC (p. 3-177) -- All variations
JMP (p. 3-358) -- All variations except m16:16 and m16:32
CMOVC (p. 3-79) -- All variations of CMOVC
Jcc (p. 3-354) -- All JNE and JNA variations (4 total)
HLT (p. 3-317) -- All variations
ROR (p. 3-660) -- All variations of ROR
SAL (p. 3-693) -- All variations of SAL
SAR (p. 3-693) -- All variations of SAR
BSWAP (p. 3-49) -- All variations
XCHG (p. 3-789) -- All variations
NOP (p. 3-508) -- All variations
PUSH (p. 3-650) -- All variations
POP (p. 3-589) -- All variations
CALL (p. 3-58) -- All variations except m16:16 and m16:32
RET (p. 3-680) -- All variations
IRET (p. 3-346) -- IRETD variation for 32-bit stack
CLD (p. 3-72) -- All variations
STD (p. 3-732) -- All variations
REP MOVS (p. 3-677) -- Repeat MOVS variation only

Prefixes

=====

Operand Size Override
Segment Register Override
Repeat String operation (for MOVS only)

Addressing Modes

=====

Immediate
Register
Mod R/M Byte
SIB Byte (except for Index==100 or Base==101)

Data Types

=====

Doubleword -- as specified by the instruction (doubleword is default)
Word -- requires operand size override
Byte -- as specified by the instruction

General Issues

=====

- * There will be no self-modifying code in the test cases.
- * There will be no illegal operands in the test cases.

Addresses

=====

- * An Effective Address is computed from the instruction, e.g., a base plus displacement specified with an r/m byte.
- * A Linear Address is calculated by summing the effective address with the appropriate segment register having been shifted 16 bits to the left for a total size of 32-bits.

Main Memory =====

- * Main memory is 32KB in size.
- * Main memory configuration is per your design.

Caches =====

- * Instruction and data cache size is limited to 1KB of data storage total.
- * Cache configuration is per your design.
- * Tag and Data storage are to be constructed from RAM parts (not SRAM).
- * In order that they be initialized to 0, valid bits may be constructed from register parts.

The Bus =====

- * A bus must connect the processor, main memory and your I/O devices.
- * The bus width and arbitration scheme are per your design.

Segmentation =====

- * Each segment has a constant (hardwired) segment limit associated with it.
- * When an effective address exceeds the segment limit a general protection exception is taken. (Not quite the full story, see Interrupts and Exceptions below.)
- * Be advised that when accessing a word or dword, all the bytes in the data must fall within the segment limit.

Virtual Memory =====

- * You are required to implement parts of a virtual memory system as specified below.
- * The page size for your project is 4KB.
- * A Translation Lookaside Buffer (TLB) will be used to translate between virtual page numbers and physical frame numbers.
- * Each TLB entry will contain at least the following flags: a valid bit, a present flag, and a read/write flag. Note a translation can be valid without the page being present in main memory, hence the valid and present flags. Depending on your design, you may or may not need a PCD flag (Page-level Cache Disable) for your I/O devices. See I/O Devices below.
- * A page can be in the "read only" state or in the "read or write state," as indicated by the read/write flag.

- * The TLB will hold 8 entries, 6 of which will be hardcoded to values specified by your TA. The other two entries are per your design, e.g., memory mapped I/O or other project specific purposes.
- * If the processor tries to write a read only page, then a general protection exception is taken.
- * If the processor tries to access a page not in the TLB or a page not present in physical memory, then a page fault is taken. (Note the abstraction: you are not required to implement the x86 page directory table for this project.)

I/O Devices =====

- * You are required to implement atleast one simple and one complicated I/O device for your project.
- * See the LC-3b documentation for simple examples of memory mapped keyboard and monitor registers.
- * More complicated I/O devices include a DMA controller.
- * Be advised that memory addresses that correspond to memory mapped I/O devices cannot be cached. You can use the TLB to indicate that a page cannot be cached by setting the PCD flag (Page-level Cache Disable) in the TLB. In this case, you will issue the request to main memory directly.

Interrupts and Exceptions =====

- * You are required to support at least one external interrupt from one of your I/O devices.
- * The two exceptions that you are required to support are: general protection exceptions and page faults. Note that both instruction and data accesses may cause these exceptions.
- * General protection exceptions are caused by writing to a read only page or when computing a memory operand with an effective address outside the CS, DS, ES, FS or GS segment limit. Note that accessing memory outside the stack segment(SS), limit causes a Stack Segment Exception which is not required for this project.
- * Page faults are caused by accessing a page that is not in the TLB or not present in physical memory.
- * There will be no nested interrupts/exceptions.
- * More detailed directions on handling interrupts and exceptions will be provided later.