Assembly Language Lecture Series: RISC-V: Load and store instructions

Roger Luis Uy

College of Computer Studies

De La Salle University

Manila, Philippines

‘2301

Load and store instructions

Load and store instructions				
LB	LBU	LH	LHU	LW
SB	SH	SW		

Load-store architecture

- RV32I is a load-store architecture, where only load and store instructions access memory and arithmetic instructions only operate on CPU registers.
- RV32I provides a 32-bit address space that is byte-addressable (2^{32}).
- Memory system can either be little-endian or big-endian (RARS use little endian)
- Instructions are stored as little-endian
- Byte = 8-bit
- Half word = 16-bit
- word = 32-bit
- memory address space is circular

Load instruction

LB *rd*, *offset*(*rs*)

- LB instruction loads an 8-bit value from memory, then sign-extends to 32 bits before storing in *rd*.
- Memory address is obtained by adding register *rs* to the sign-extended 12-bit *offset*.
- pseudo-instruction *la* (load address) is used to initialize a register to point to a memory

la x5, var1

LB x10, 0(x5)

x5 = 10010000

memory address:

0+10010000 = 10010000

x10 = FFFFFF88

Label	Address (hex)	Memory data (hex)
	1001000F	00
	1001000E	00
	1001000D	00
var4	1001000C	FF
	1001000B	CC
	1001000A	DD
	10010009	EE
var3	10010008	FF
	10010007	81
	10010006	AB
	10010005	CD
var2	10010004	EF
	10010003	BB
	10010002	AA
	10010001	99
var1	10010000	88

Load instruction

LBU *rd*, *offset*(*rs*)

- LBU instruction loads an 8-bit value from memory, then zero-extends to 32 bits before storing in *rd*.
- Memory address is obtained by adding register *rs* to the sign-extended 12-bit *offset*.
- pseudo-instruction *la* (load address) is used to initialize a register to point to a memory

la x5, var1

LBU x10, 0(x5)

x5 = 10010000

memory address:

0+10010000 = 10010000

x10 = 00000088

Label	Address (hex)	Memory data (hex)
	1001000F	00
	1001000E	00
	1001000D	00
var4	1001000C	FF
	1001000B	CC
	1001000A	DD
	10010009	EE
var3	10010008	FF
	10010007	81
	10010006	AB
	10010005	CD
var2	10010004	EF
	10010003	BB
	10010002	AA
	10010001	99
var1	10010000	88

Load instruction

LH *rd*, *offset*(*rs*)

- LH instruction loads a 16-bit value from memory, then sign-extends to 32 bits before storing in *rd*.
- Memory address is obtained by adding register *rs* to the sign-extended 12-bit *offset*.
- pseudo-instruction *la* (load address) is used to initialize a register to point to a memory

la x5, var1

LH x10, 0(x5)

x5 = 10010000

memory address:

0+10010000 = 10010000

x10 = FFFF9988

Label	Address (hex)	Memory data (hex)
	1001000F	00
	1001000E	00
	1001000D	00
var4	1001000C	FF
	1001000B	CC
	1001000A	DD
	10010009	EE
var3	10010008	FF
	10010007	81
	10010006	AB
	10010005	CD
var2	10010004	EF
	10010003	BB
	10010002	AA
	10010001	99
var1	10010000	88

Load instruction

LHU *rd*, *offset*(*rs*)

- LHU instruction loads a 16-bit value from memory, then zero-extends to 32 bits before storing in *rd*.
- Memory address is obtained by adding register *rs* to the sign-extended 12-bit *offset*.
- pseudo-instruction *la* (load address) is used to initialize a register to point to a memory

la x5, var1

LHU x10, 0(x5)

x5 = 10010000

memory address:

0+10010000 = 10010000

x10 = 00009988

Label	Address (hex)	Memory data (hex)
	1001000F	00
	1001000E	00
	1001000D	00
var4	1001000C	FF
	1001000B	CC
	1001000A	DD
	10010009	EE
var3	10010008	FF
	10010007	81
	10010006	AB
	10010005	CD
var2	10010004	EF
	10010003	BB
	10010002	AA
	10010001	99
var1	10010000	88

Load instruction

LW *rd*, *offset*(*rs*)

- LW instruction loads a 32-bit value from memory to *rd*.
- Memory address is obtained by adding register *rs* to the sign-extended 12-bit *offset*.
- pseudo-instruction *la* (load address) is used to initialize a register to point to a memory

la x5, var1

LW x10, 0(x5)

x5 = 10010000

memory address:

0+10010000 = 10010000

x10 = BBAA9988

Label	Address (hex)	Memory data (hex)
	1001000F	00
	1001000E	00
	1001000D	00
var4	1001000C	FF
	1001000B	CC
	1001000A	DD
	10010009	EE
var3	10010008	FF
	10010007	81
	10010006	AB
	10010005	CD
var2	10010004	EF
	10010003	BB
	10010002	AA
	10010001	99
var1	10010000	88

Store instruction

SB *rs2*, *offset*(*rs1*)

- SB instruction stores an 8-bit value from the LSB of register *rs*2 to memory.
- Memory address is obtained by adding register *rs* to the sign-extended 12-bit *offset*.

la x5, var1

la x6, var2

LW x10, 0(x5)

SB x10, (x6)

x6 = 10010004

memory address:

0+10010004 = 10010004

x10 = BBAA9988

Label	Address (hex)	Memory data (hex)
	1001000F	00
	1001000E	00
	1001000D	00
var4	1001000C	FF
	1001000B	CC
	1001000A	DD
	10010009	EE
var3	10010008	FF
	10010007	81
	10010006	AB
	10010005	CD
var2	10010004	~~EF~~ 88
	10010003	BB
	10010002	AA
	10010001	99
var1	10010000	88

Store instruction

SH *rs2*, *offset*(*rs1*)

- SH instruction stores a 16-bit value from the least significant half-word of register *rs*2 to memory.
- Memory address is obtained by adding register *rs* to the sign-extended 12-bit *offset*.

la x5, var1

la x6, var2

LW x10, 0(x5)

SH x10, (x6)

x6 = 10010004

memory address:

0+10010004 = 10010004

x10 = BBAA9988

Label	Address (hex)	Memory data (hex)
	1001000F	00
	1001000E	00
	1001000D	00
var4	1001000C	FF
	1001000B	CC
	1001000A	DD
	10010009	EE
var3	10010008	FF
	10010007	81
	10010006	AB
	10010005	~~CD~~ 99
var2	10010004	~~EF~~ 88
	10010003	BB
	10010002	AA
	10010001	99
var1	10010000	88

Store instruction

SW *rs2*, *offset*(*rs1*)

- SW instruction stores a 32-bit value of register *rs*2 to memory.
- Memory address is obtained by adding register *rs* to the sign-extended 12-bit *offset*.

la x5, var1

la x6, var2

LW x10, 0(x5)

SW x10, (x6)

x6 = 10010004

memory address:

0+10010004 = 10010004

x10 = BBAA9988

Label	Address (hex)	Memory data (hex)
	1001000F	00
	1001000E	00
	1001000D	00
var4	1001000C	FF
	1001000B	CC
	1001000A	DD
	10010009	EE
var3	10010008	FF
	10010007	~~81~~ BB
	10010006	~~AB~~ AA
	10010005	~~CD~~ 99
var2	10010004	~~EF~~ 88
	10010003	BB
	10010002	AA
	10010001	99
var1	10010000	88