AUBIE CPU SPECIFICATION

DATAPATH ELEMENTS

Memory
The memory of the Aubie is 32-bits wide (i.e. a dlx_word is stored at every address). Can do one read or one write per cycle. Note: Bit 31 is the most significant bit, and bit 0 is the least significant bit of a memory word.

ALU
See assignment 3.

Registers
32 registers, single ported, can do one read or one write per cycle

2-way mux
controlled by a one bit signal from controller

3-way mux
controlled by a two-bit signal from controller

PC incrementer
Combinational logic that adds one (unsigned) to its' (32 bit) input

INSTRUCTION SET

The opcode is always found in the high order 8 bits of an instruction word (bits 31 to 24) . Some instructions are one dlx-word long, and some are two words long. In addition to the opcode, the first word may hold a number of register numbers for source and destination registers. The second word contains either a 32-bit immediate value for the load immediate (LDI) instruction or it holds an address for the store (STO),load (LD), jump (JMP), or jump-if-zero (JZ) instructions.

ALU INSTRUCTIONS
1 dlx-word long (1 address)

FORMAT , word 1

Opcode	**Dest**	**Op1**	**Op2**	**Not used**
Bits 31-24	Bits 23-19	Bits 18-14	Bits 13-9	Bits 8-0

Perform the operation on registers **op1** and **op2** and put the result in register **dest**.

OPCODES

Mnemonic	**Opcode**	**Meaning**
ADDU dest, op1, op2	0x00	unsigned add
SUBU dest, op1, op2	0x01	unsigned subtract

ADD	dest,op1,op2	0x02	two's complement add
SUB	dest,op1,op2	0x03	two's complement subtract
MUL	dest,op1,op2	0x04	two's complement multiply
DIV	dest,op1,op2	0x05	two's complement divide
ANDL	dest,op1,op2	0x06	logical and
ANDB	dest,op1,op2	0x07	bitwise and
ORL	dest,op1,op2	0x08	logical or
ORB	dest,op1,op2	0x09	bitwise or
NOTL	dest,op1,op2	0x0a	logical not (op1)
NOTB	dest,op1,op2	0x0b	bitwise not (op1)

Examples

ADDU R1,R2,R3 encodes as 0x00088300
NOTB R7,R8,R9 encodes as 0x0b3a1200

STORE INSTRUCTIONS

2 dlx words long, stored in 2 consecutive addresses in memory

FORMAT, word 1

Opcode	Dest not used	Op1	Op2 not used	Not used
Bits 31-24	Bits 23-9	Bits 18-14	Bits 13-9	Bits 8-0

Word 2

Address
Bits 31-0

STO op1,address 0x20 put contents of reg op1 in memory specified by address (word 2 of instruction)

Example

STO R2,0x12341234 encodes as 0x20008000, 0x12341234

LOAD INSTRUCTIONS

2 dlx words long, stored in 2 consecutive addresses in memory

FORMAT, word 1

Opcode	Dest	Op1 not used	Op2 not used	Not used
Bits 31-24	Bits 23-19	Bits 18-14	Bits 13-9	Bits 8-0

Word 2

Address or Immediate
Bits 31-0

LD dest, addr 0x30 load contents of addr to

		register dest
LDI dest,#imm	0x31	load value imm into reg. dest

Examples
LD R5,0x00004412 encodes as 0x30280000, 0x00004412
LDI R1,#0x12121212 encodes as 0x31080000, 0x12121212

REGISTER INDIRECT LOAD AND STORE
These do load and store using the contents of a register to specify the address. For STOR, the dest register holds the address to which to store the contents of reg. op1 . For LDR, the op1 reg holds the address to load the contents from into the dest register

1 dlx word long
Format, word 1

Opcode	**Dest**	**Op1**	**Op2 not used**	**Not used**
Bits 31-24	Bits 23-19	Bits 18-14	Bits 13-9	Bits 8-0

STOR (dest), op1	0x22	put contents of reg op1 in address given by contents of dest reg
LDR dest, (op1)	0x32	load contents of addr given by register op1 into reg. dest

Examples
STOR (R7),R8 encodes as 0x223a0000
LDR R11,(R12) encodes as 0x325b0000

JUMP OPERATIONS
Either unconditional (JMP) or condition (JZ) jump to an address given in the 2nd word of the instruction

2 dlx words long, stored in two consecutive addresses in memory
Format, word 1

Opcode	**Dest not used**	**Op1**	**Op2 not used**	**Not used**
Bits 31-24	Bits 23-19	Bits 18-14	Bits 13-9	Bits 8-0

Word 2

Address
Bits 31-0

JMP addr 0x40 unconditional jump to addr
JZ op1, addr 0x41 jump to addr if op1 == 0

Examples
JMP 0x11111111 encodes 0x40000000,0x111111111
JZ R7,0x22220000 encodes as 0x41380000 ,0x22220000

MISC OPERATIONS

No Operation, 1 dlx word long

Opcode	**Not Used**
Bits 31-24	Bits 23-0

NOOP 0x10 do nothing at all

Example
NOOP encodes as 0x10000000