Computer Aided Design Project (CAP)

Project title: Design and implement “CAP17” processor using **pipelined** style
Department: Computer Engineering Department, Iran university of Science and Technology

You should send your Verilog codes and also simulation results .

Project description

CAP17 CPU is a 16 bit-RISC processor. CAP17 processor has 5 set of instructions.

- The data transfer-reference set of instructions such as load, store instructions and etc.
- The arithmetic-reference set of instruction such as add, subtract and etc.
- The logical-reference set of instruction such as and, or and etc.
- The conditional branch-reference set of instruction.
- The unconditional jump-reference set of instruction.

1. Register sets

CAP17 has 14 registers. Registers in CAP17 are 16 bits. Each word in CAP17 has 16 bits. Table 1 shows CAP17’s registers and their description.

Tabel 1: CAP17 processor registers and their description

Data Registers	D0 ~ D3	4 fast registers to perform arithmetic
Address Registers	A0 ~ A3	4 register to access the memory
Base Address	BA	Base address
Program counter	PC	Program counter
Status register	SR	Status register
Zero register	Zero	Always equal to the Zero
High register	Hi	Contain high part (16 bit) of 32 bit multiply product
Low register	Lo	Contain low part (16 bit) of 32 bit multiply product

2. Status register

Status register (flag register) like as other registers is a 16-bit register, but you just need the first four bits of it (figure 1).

Z = 1 if result =0;
N = 1 if result<0;
C = 1 if result has carry;
O = 1 if result has overflow;

Figure 1: Status register

3. Assembly language of CAP17 processor

As it has been mentioned above CAP17 has five assembly categories. Table 2 shows CAP17 assembly language instruction set within their description and also an example for each of them.

Table 2: CAP17 processor assembly language instruction set

Category	Instruction	Example	Meaning	comments
Arithmetic	add	Add d0, d1	d0= d0 + d1	Two operands; overflow detected (opcode =0000)
	add	Add d0, 25	d0 = d0 + memory(BA+50)	Two operands; overflow detected (opcode = 0001)
	subtract	Sub d0, d1	d0 = d0 – d1	Two operands; overflow detected (opcode = 0010)
	Add immediate	Addi d0, 50	d0 = d0 + 50	+ constant; overflow detected (opcode = 0011)
	Multiply	Mul d0, d1	Hi, Lo = d0 × d1	32 bit signed product in Hi, Lo (opcode = 0100)
Logical	And	And d0, d1	d0 = d0 & d1	Two reg. operand; logical and (opcode = 0101)
	Shift left logical	Sll d0, 10	d0 = d0<<10	Shift left by constant (opcode = 0110)
Data Transfer	Load word	Lw d0, 7	d0 = memory(BA + 14)	Word from memory to register (14 = 7×2) (opcode = 0111)
	Load indirect word	Lwi d0,7	d0 = mem(mem(BA+14))	Word from memory of memory is moved to reg (opcode = 1000)
	Store word	Sw d0, 10	Memory(BA+ 20) = d0	Word from register to memory (20= 10×2) (opcode = 1001)
	Store indirect word	Swi d0,10	Mem(Mem(BA + 20)) = d0	Reg is store in memory of memory location (opcode = 1010)
	Clear reg or mem	CLR d0	d0 = 0	One operand; clear reg or clear memory (opcode = 1011)
	Move immediate	Mov BA, 50	50 → BA	Move immediate to BA reg (opcode = 1100)

Compare and conditional branch	compare	CMP d0, d1	If (d0-d1=0)
			Z flag = 1
			Elsif (d0 – d1<0)
			N flag = 1
			Else
			Z flag and N flag are zero
				It does not change content of operands. CMP instruction change content of status register (flags) (opcode = 1101)

	Branch on not equal	Bne 25	PC = PC + 2 + 50	Bne checks Z flag and if Z flag is zero, it go to location (25 × 2); PC relative (opcode = 1110)
Unconditional Jump	Simple jump	Jmp 2500	Go to 5000	Jump to target address (2500 × 2) (opcode = 1111)

The multiply product result is stored in the following two registers (Hi and Lo)

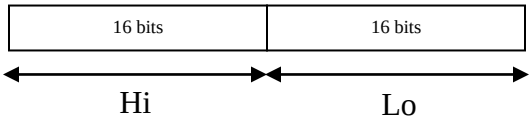

Figure 2: Multiply product result

4. Memory access in CAP17

As any word in CAP17 has 2 bytes (16 bits), words must always start at addresses that are multiply of 2. This requirement is called an alignment restriction, and many architectures have it. This type addressing is called byte addressing. Figure 3 shows CAP17 memory addresses. For example the address of third word is 8.

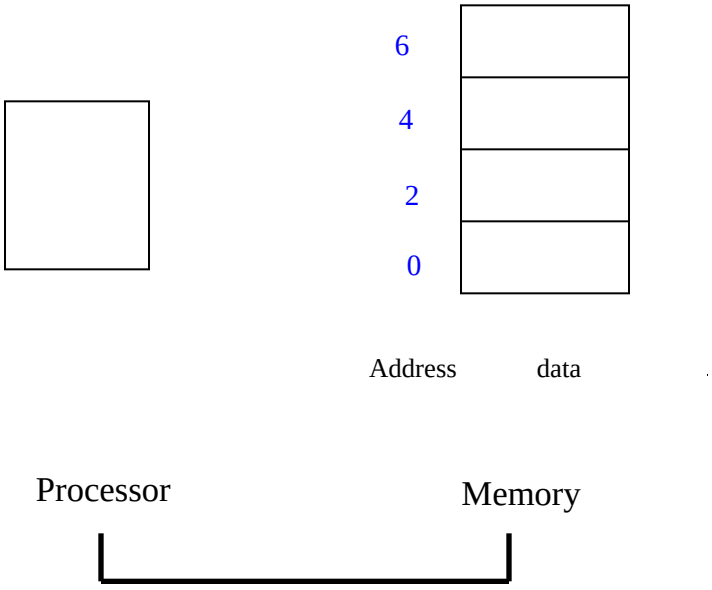

Figure 3: byte addressing in CAP17 processor

5. Types of memory in CAP17

The CAP17 processor has unique memory for both instruction memory and data memory. The CAP17 processor has 4 KB memory. The first 1 KB is used for instruction memory; while the rest of 3 KB memory is used for data memory.
As there is one memory for instruction and data memory, the structural hazard is obvious.

6. Machine code in CAP17 processor

CAP17 has two formats (R-format and I format) for machine code of instructions as follows

6.1. R-format

R-format (Register format) as the follows (figure 4):

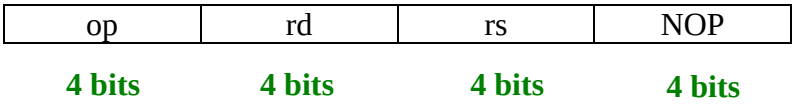

Figure 4: R-format

- op: Basic operation of the instruction, traditionally called *opcode*.
- rd: The destination register
- rs: The source register
- NOP: no operational code (don't care)

6.2 I-format

The I-format as follows (figure 5):

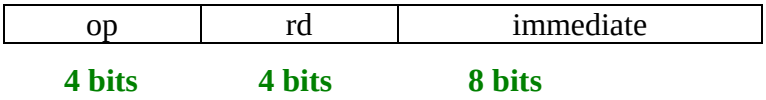

Figure 5: I-format

6.3 J-format

The J-format (jump format) as follows (figure 6):

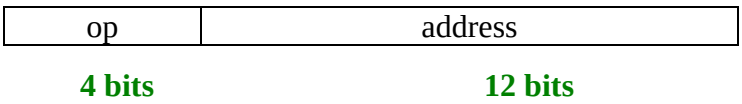

Figure 6: J-format

Table 3 shows the register convention for the CAP17 assembly language.

Table 3: register numbers in CAP17 processor

Name	Register number	usage
Zero	0	The constant value 0
D0 ~ D3	1-4	Data registers
A0 ~ A3	5-8	Address register
SR	9	Status register
BA	10	Base Address
PC	11	Program Counter
Hi	12	High part for multiply
Lo	13	Low part for multiply

7. Hazard solution

You are free to handle hazard in any situation you like (Additional Score)

8. Input and Output of your program

Input1: A complete Test Fixture written by yourself, to cover all parts of the CPU.

Input2: your CPU has 2 inputs a) clock and b) reset

Output: All important signals (Address of mem, data of mem, PC, status reg, ALU output, etc)

Success!!!