Assignment 4

1. Let the address stored in the program counter be designated by the symbol X1. The instruction stored in X1 has an address part (operand reference) X2. The operand needed to execute the instruction is stored in the memory word with address X3. An index register contains the value X4. What is the relationship between these various quantities if the addressing mode of the instruction is (a) direct; (b) indirect; (c) PC relative; (d) indexed?

Answer:

a.
X3=X2=(X1)

b.
X2=(X1)
X3=(X2)

c.
X3=X1+X2+1

d.
X3=X2+X4

2. A nonpipelined processor has a clock rate of 2.5 GHz and an average CPI (cycles per instruction) of 4. An upgrade to the processor introduces a five-stage pipeline. However, due to internal pipeline delays, such as latch delay, the clock rate of the new processor has to be reduced to 2 GHz.
a. Assume a program has 100 instructions, then what is the speedup achieved for this program?
b. What is the MIPS rate for each processor?

Answer:

a.

$$S_k = \frac{nk}{k+(n-1)} = \frac{5*100}{5+(100-1)} = 4.8$$
$$S = S_k * \frac{2GHz}{2.5GHz} = 4.8 * 0.8 = 3.8$$

b.
MIPS(Million Instructions Per Second)

For the original processer, it is a nonpipelined processor. It has frequency of 2.5GHz which means 2.5G cycles per second, which is 2500M cycles per second, and each instruction requires 4 clock cycles. So the MIPS rate is 2500/4= 625 MIPS

For the upgraded processer, it has a five-stage pipeline, meaning each instruction only takes one cycle. So the MIPS rate is 2000 MIPS

3. A RISC machine's compiler may do both a mapping of symbolic registers to actual registers and a rearrangement of instructions for pipeline efficiency. An interesting question arises as to the order in which these two operations should be done. Consider the following program fragment:

```
LD SR1, A               ;load A into symbolic register 1
LD SR2, B               ;load B into symbolic register 2
ADD SR3, SR1, SR2       ;add contents of SR1 and SR2 and store in SR3
LD SR4, C
LD SR5, D
ADD SR6, SR4, SR5
```

a. First do the register mapping and then any possible instruction reordering. How many machine registers are used? Has there been any pipeline improvement?
b. Starting with the original program, now do instruction reordering and then any possible mapping. How many machine registers are used? Has there been any pipeline improvement?

Answer:

a.
register mapping:

```
LD R1,A
LD R2,B
ADD R1,R1,R2
LD R2,C
LD R3,D
ADD R2,R2,R3
```

Three registers are used, no pipeline improvement for they both needs Register 2

b.
instruction reordering:

```
LD SR1, A
LD SR2, B
LD SR4, C
LD SR5, D
ADD SR3, SR1, SR2
ADD SR6, SR4, SR5
```

register mapping:

```
LD R1, A
LD R2, B
LD R3, C
LD R4, D
ADD R5, R1, R2
ADD R1, R3, R4
```

Five register are used, have pipeline improvement.

4. Figure 1 shows an example of a superscalar processor organization. The processor can issue two instructions per cycle if there is no resource conflict and no data dependence problem. There are essentially two pipelines, with four processing stages (fetch, decode, execute, and store). Each pipeline has its own fetch decode and store unit. Four functional units (multiplier, adder, logic unit, and load unit) are available for use in the execute stage and are shared by the two pipelines on a dynamic basis. The two store units can be dynamically used by the two pipelines, depending on availability at a particular cycle. There is a lookahead window with its own fetch and decoding logic. This window is used for instruction lookahead for out-of-order instruction issue. Consider the following program to be executed on this processor:

```
I1: Load R1, A /R1 ← Memory (A)/
I2: Add R2, R1 /R2 ← (R2) + R1/
I3: Add R3, R4 /R3 ← (R3) + R4/
I4: Mul R4, R5 /R4 ← (R4) x R5/
I5: Comp R6     /R6 ← (R6)/
I6: Mul R6, R7 /R6 ← (R6) x R7/
```

Show the pipeline activity for this program on the processor of Figure 1 using out-order issue with out-order completion policies

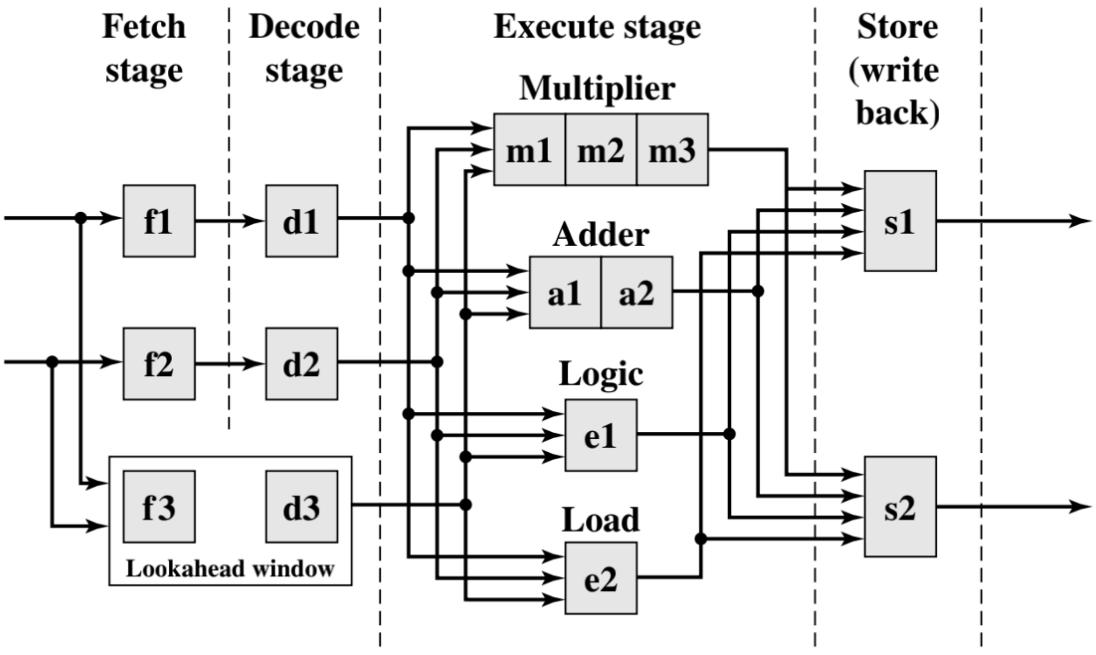

Figure 1. A Dual-Pipeline Superscalar Processor

Answer:

I3	f1	d1	a1	a2	s1		
I4	f2	d2	m1	m2	m3	s2	
I5	f3	d3	e1	s1			
	I6	f1	d1	m1	m2	m3	s2
	I1	f2	d2	e2	s2		
		I2	f1	d1	a1	a2	s1