1.

Stage	Leave
Fetch	icode: ifun ← M1[PC] valP ← PC+1
Decode	valA ← R[%ebp] valB ← R[%ebp]
Execute	valE ← valB+4
Memory	valM ← M4[valA]
Write Back	R[%esp] ← valE R[%ebp] ← valM
Update PC	PC ← valP

2. There is a load/use hazards between mrmovl(2) and mrmovl(3) and a load/use hazards between mrmovl(3) and addl(4). Data hazard between addl(1) and mrmol(2) can be solved by forwarding.

3.

About the graph, 4 registers inserted after G and between A and B, C and D, D and E, the forth place is optional.

Throughput = 1/90ps * 1000= 11.11 GOPS.

Latency = 90ps * 4 = 360 ps.

4.

bool instr_valid = icode in {INOP, IHALT, IRRMOVL, IIRMOVL, IMRMOVL, IOPL, IJXX, ICALL, IRET, IPUSHL, IPOPL};

5.

This code is similar to the code for srcA.

```
int srcB = [
        icode in { IOPL, IRMMOVL, IMRMOVL } : rB;
        icode in { IPUSHL, IPOPL, ICALL, IRET } : RESP;
        1 : RNONE; # Don’t need register
];
```