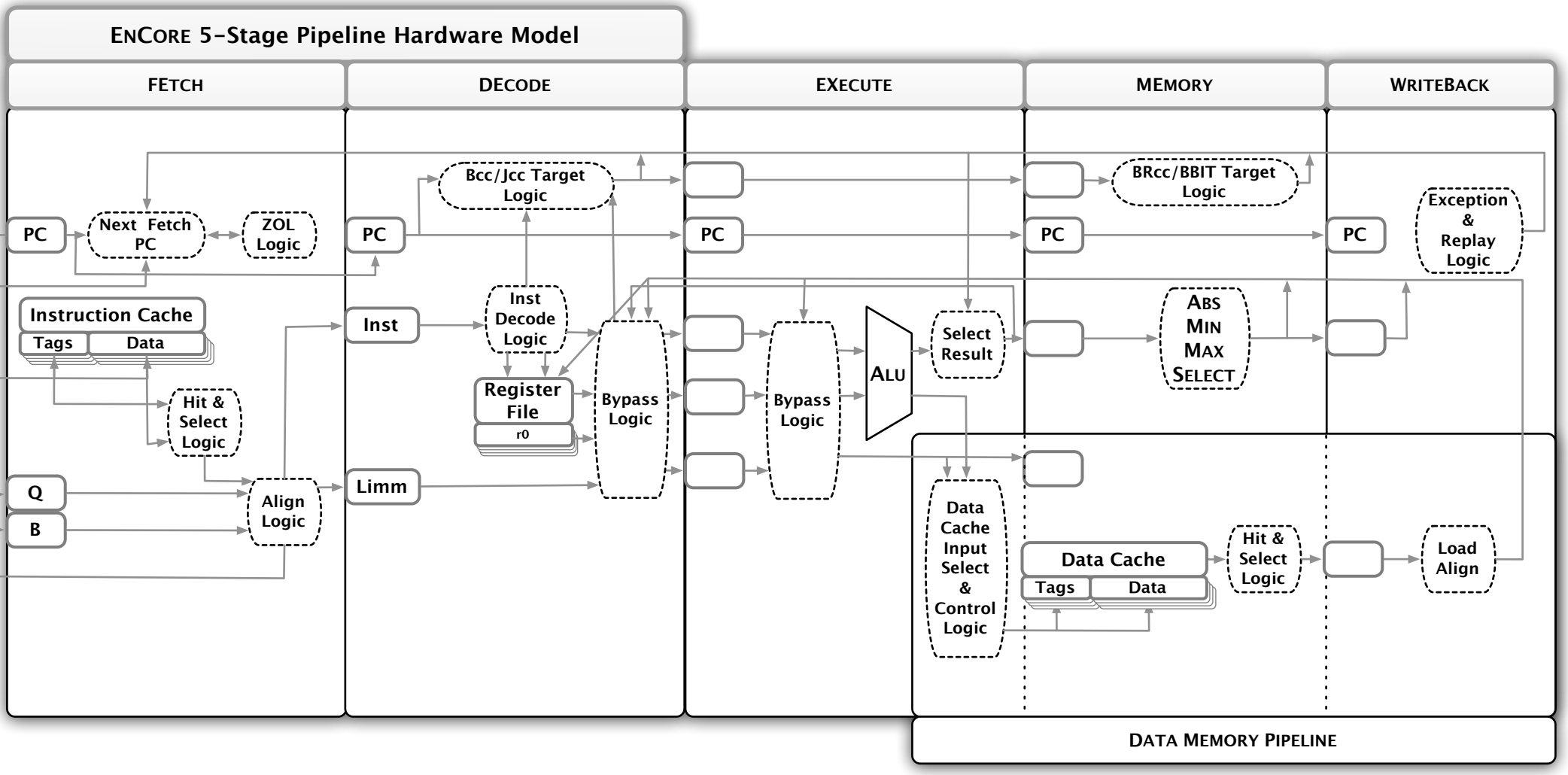

EnCore 5-Stage Pipeline JIT Generated Software Model

FETCH
DECODE
EXECUTE
MEMORY
WRITEBACK

```
void
pipeline(uint64_t  opd1,  uint64_t  opd2,
         uint64_t* dst1,  uint64_t* dst2,
         uint32_t  faddr, uint32_t  xc,   uint32_t mc)
{
  // FETCH     - account for instruction fetch latency                          1
  cpu.pl[FE] += fetch(faddr);
  // INVARIANT - see section 3.1 processor pipeline model                       2
  if (cpu.pl[FE] < cpu.pl[DE]) cpu.pl[FE] = cpu.pl[DE];

  // DECODE    - determine operand availability time                            3
  cpu.pl[DE] = max3((cpu.pl[FE] + 1), opd1, opd2);
  if (cpu.pl[DE] < cpu.pl[EX]) cpu.pl[DE] = cpu.pl[EX];

  // EXECUTE   - account for execution latency and destination availability time 4
  cpu.pl[EX] = *dst1 = cpu.pl[DE] + xc;
  if (cpu.pl[EX] < cpu.pl[ME]) cpu.pl[EX] = cpu.pl[ME];

  // MEMORY    - account for memory latency and destination availability time   5
  cpu.pl[ME] = *dst2 = cpu.pl[EX] + mc;
  if (cpu.pl[ME] < cpu.pl[WB]) cpu.pl[ME] = cpu.pl[WB];

  // WRITEBACK
  cpu.pl[WB] = cpu.pl[ME] + 1;
}
```