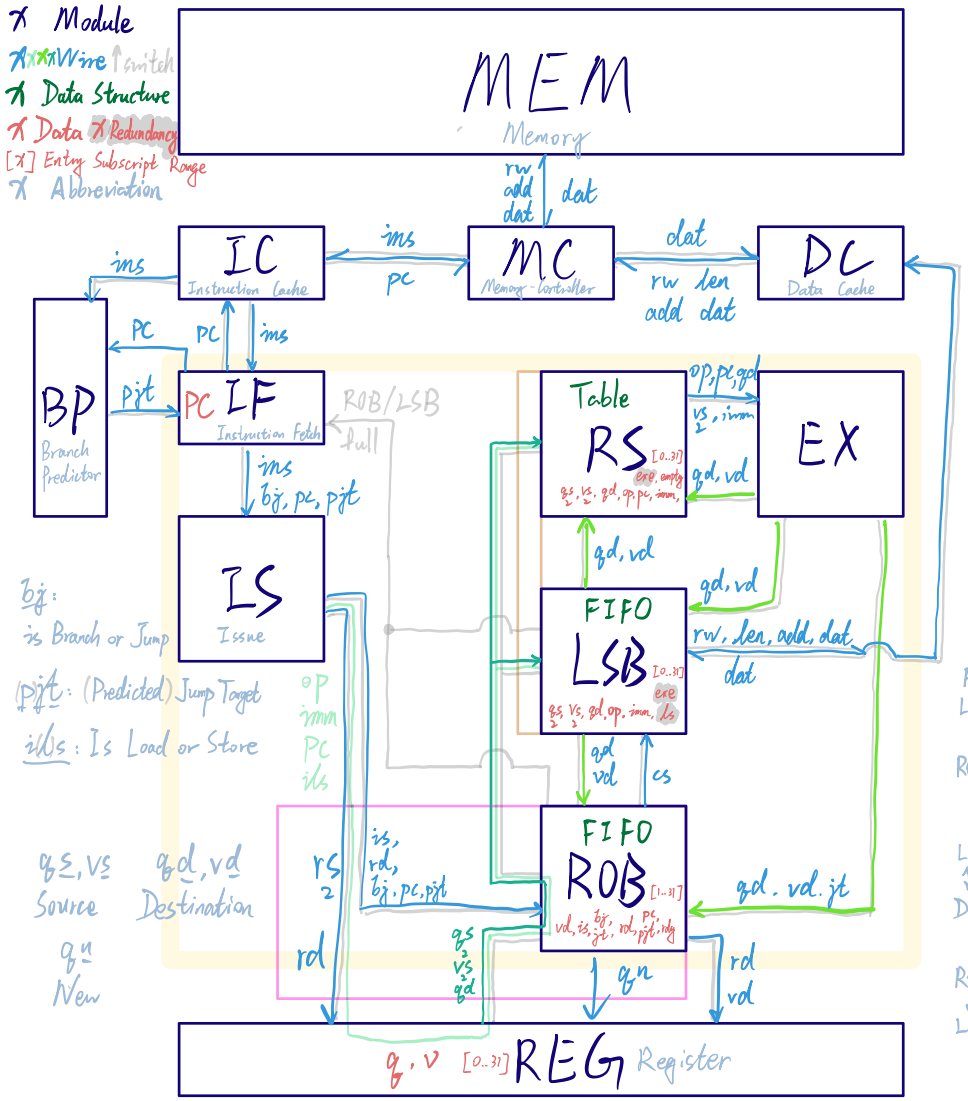

RS: Reserve Station
ROB: Reordered Buffer
LSB: Load-Store Buffer
EX: EXecution

RS & LSB input: op, imm, vs, qs, qd (PC ← ROB)

LSB & EX output: qd, vd ↑ jt ↓ to ROB, RS, LSB

RS / LSB: exe: EXEcutable; ls: Load or Store
ROB: rdy: ReaDY to be committed
LSB ⇕ DC: rw: Read or Write; len: LENgth; addr: ADDress; dat: DATa
ROB ↓ LSB: cs: Commit Store

IS. vs & rd → REG
REG[rd].q = qn
REG[rs].vs & qs
ROB[qn].pc & rd ← IS.pc & rd
ROB.qs = REG.qs
ROB.vs = (ROB[qs].en) ? (ROB[qs].v) : (REG.vs)

LSB 内依次执行
ROB 内依次 Commit
ROB 为 Branch/Jump 指令精确中断:
1. Store 指令完成后对应 entry in ROB 才能 Commit
2. 预测失败则清空
IF, IS, ROB, RS, EX, LSB 以及 REG 中的 Q
保留前缀 Committed STORE