A Simplified RISCV-CPU Design

```mermaid graph LR A[Christmas] -->|Get money| B(Go shopping) B --> C{Let me think} C -->|One| D[Laptop] C -->|Two| E[iPhone] C -->|Three| F[fa:fa-car Car] C -->A ``` ```mermaid %%{ init: { 'flowchart': { 'curve': 'linear' } } }%% graph LR classDef Memory fill:#f9f,stroke:#333,stroke-width:3px; classDef Process fill:#f3f,stroke:#111,stroke-width:3px; Mem[fa:fa-memory 128KB Memory] IC(fa:fa-circle-info L1-ICache) DC(fa:fa-database L1-Dcache) Reg(Register) subgraph Fetcher PC(PC reg) Inst(Instruction) end subgraph Dispatcher decode(Decoder) Process(Processer) end LSB(LoadStoreBuffer) RS(ReservationStation) ALU(ALU) SLU(SLU) BUS(BUS) ROB(ReorderBuffer) Mem -->|Fetch| IC IC -->|Write?| Mem Mem -->|Load| DC DC -->|Write Back| Mem IC -->|Read| PC IC --> Inst Inst--> decode decode--> Process Process-->|dispatch|RS Process-->|dispatch|LSB Reg-->|Decoded v_j,v_k|Process RS-->|Issue|ALU LSB-->|Issue|SLU DC-->|Data|SLU ALU-->ROB SLU-->ROB ALU-->BUS SLU-->BUS BUS-->|Update qj,qk|RS ROB-->|Commit|Reg zROB-->|Commit|DC class A1 Memory; ```
```