

The diagram illustrates the internal architecture of a computer, showing the flow of data and instructions between various components.

**Data Bus (DBUS):** Connects the ALU, registers, cache, and control units.

**Instruction Bus (IBUS):** Connects the instruction cache, program counter, and instruction decoder.

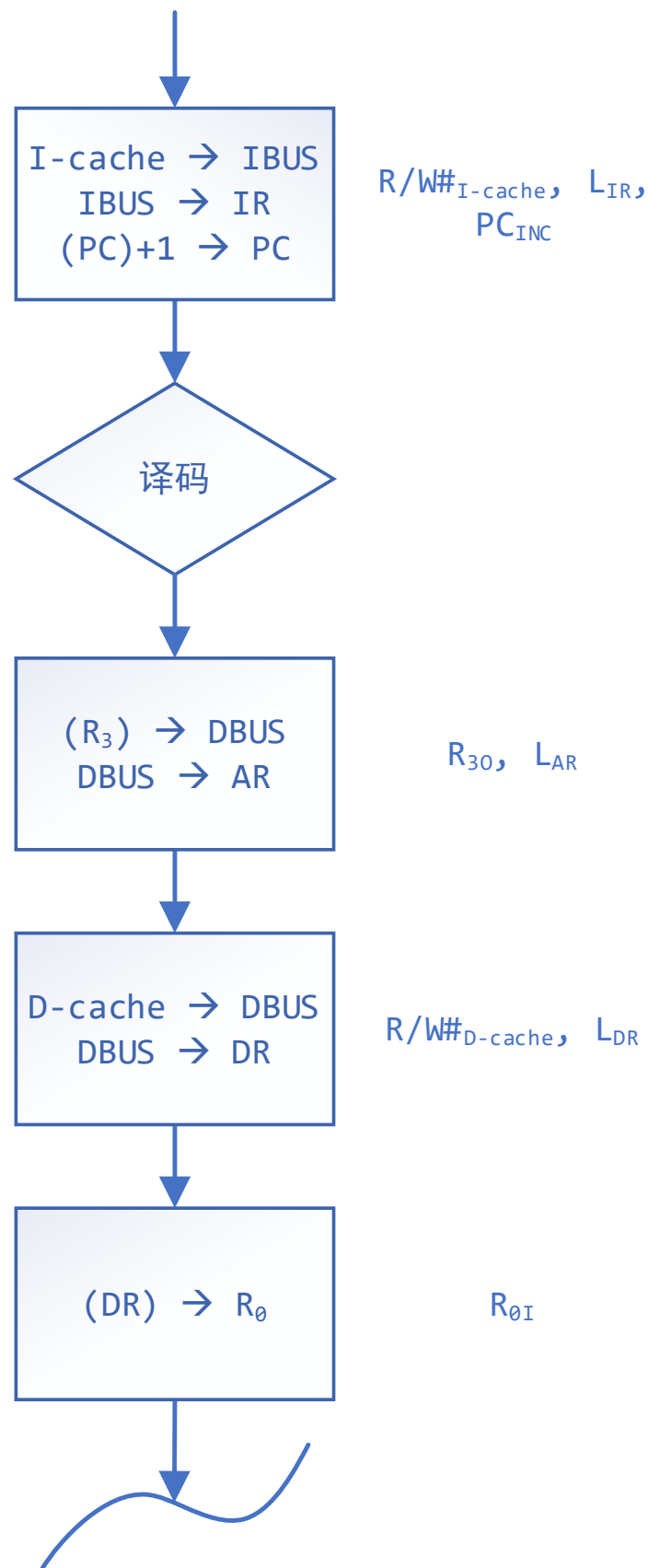
**Internal Components:**

- ALU (Arithmetic Logic Unit):** Performs arithmetic and logical operations. It receives data from the registers and the data bus, and sends results back to the registers and the data bus.
- Registers:**
  - 通用寄存器 (General Registers):** A set of registers (R0, R1, R2, R3) used for storing data and addresses.
  - 数据缓冲寄存器 (Data Buffer Register):** A register used for buffering data between the ALU and the data bus.
  - 状态字寄存器 (Status Word Register):** A register used for storing the status of the processor (e.g., flags).
- Cache:**
  - 数据缓存 (Data Cache):** A cache for storing data. It contains a table of addresses and data.
  - 指令缓存 (Instruction Cache):** A cache for storing instructions. It contains a table of addresses, operation codes, and address codes.
- Control Units:**
  - 时序发生器/操作控制器 (Timing Generator/Operation Controller):** Generates control signals (C1, C2, ..., Cn) for the ALU, registers, and cache.
  - 指令译码器 (Instruction Decoder):** Decodes instructions from the instruction bus and sends control signals to the ALU and registers.
- Program Counter (PC):** Holds the address of the next instruction to be executed.
- Address Register (AR):** Holds the address of the data being accessed in the data cache.
- Instruction Register (IR):** Holds the instruction being executed.

**Data Flow:**

- Data is fetched from the data cache to the ALU via the DBUS.
- Instructions are fetched from the instruction cache to the instruction register via the IBUS.
- The instruction register sends the instruction to the instruction decoder, which then sends control signals to the ALU and registers.
- The ALU sends results back to the registers and the data bus.
- The registers send data back to the ALU and the data bus.
- The program counter sends the address of the next instruction to the instruction cache via the IBUS.
- The address register sends the address of the data being accessed to the data cache via the DBUS.

取数指令  $\boxed{\text{LAD } (R_3), R_0}$  的指令周期流程图绘制如下, 其中  $L_{IR}$ ,  $L_{AR}$ ,  $L_{DR}$  分别表示 IR, AR, DR 等寄存器的载入信号,  $R_{i/I/O}$  表示  $R_i$  的输入输出使能信号,  $R/W\#_{X\text{-cache}}$  表示 X-cache 的读入写出信号。



2、假设某机器有 80 条指令，平均每条指令由 4 条微指令组成，其中有一条取指微指令是所有指令公用的。已知微指令长度为 32 位，请估算控制存储器容量。

**解答：**

微指令共  $3 \times 80 + 1 = 241$  条，从而控制存储器容量

$$M = 241 \times \frac{32}{8} \text{ Byte} = 964 \text{ Byte}.$$

3、某计算机有如下部件：ALU，移位器，主存 M，主存数据寄存器 MDR，主存地址寄存器 MAR，指令寄存器 IR，通用寄存器  $R_0 \sim R_3$ ，暂存器 C 和 D。

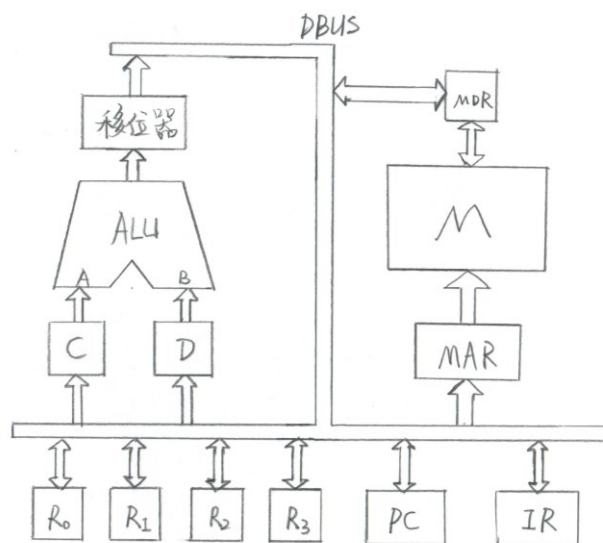
(1) 请将各逻辑部件组成一个数据通路，并标明数据流动方向。

(2) 画出 **ADD  $R_1, R_2$**  指令的指令周期流程图

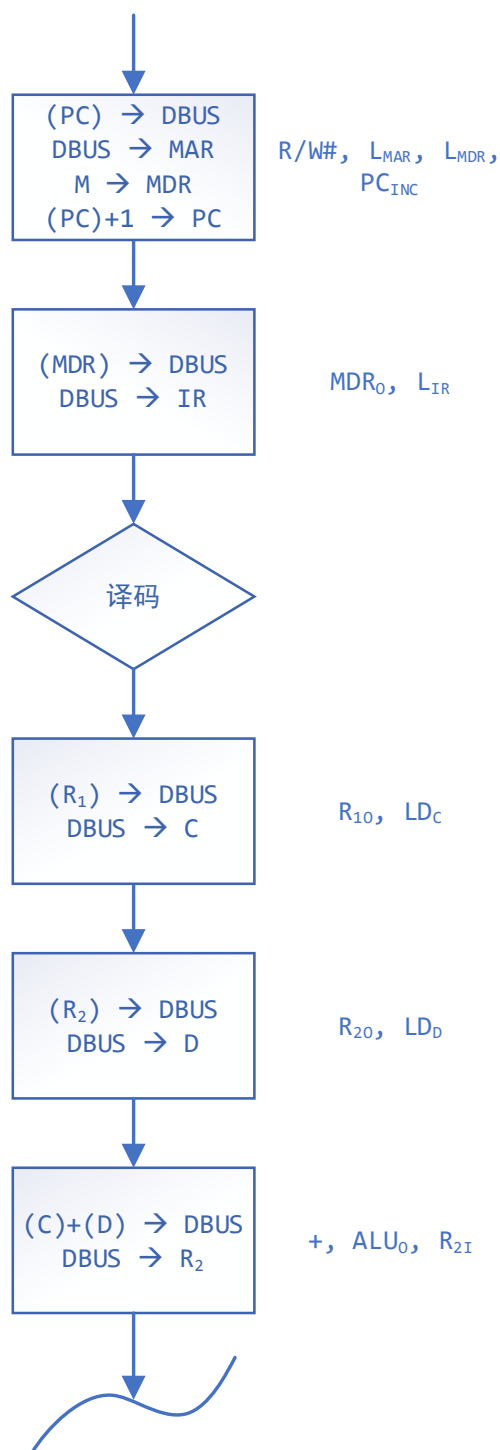
$$(R_1) + (R_2) \rightarrow R_2.$$

**解答：**

(1) 构建数据通路如图，可行的数据流动方向见总线上箭头的方向。



(2) **ADD R<sub>1</sub>, R<sub>2</sub>** 指令的指令周期流程图如下, 其中 R/W#表示主存的读写信号, L<sub>MAR</sub>, L<sub>MDR</sub>, L<sub>IR</sub>, LD<sub>C</sub>, LD<sub>D</sub> 分别表示寄存器 MAR, MDR, IR 和暂存器 C, D 的载入信号。R<sub>1O</sub>, R<sub>2O</sub> 分别表示 R<sub>1</sub>, R<sub>2</sub> 的输出使能信号, R<sub>2I</sub> 表示 R<sub>2</sub> 的写入使能信号。



4、已知某机采用微程序控制方式，控存容量为  $512 \times 48$  位。微程序可在整个控存中实现转移，控制微程序转移的条件共 4 个，微指令采用水平型格式，后继微指令地址采用断定方式。请问：

- (1) 微指令的三个字段分别应为多少位？
- (2) 画出对应这种微指令格式的微程序控制器逻辑框图。

**解答：**

(1) 判别测试字段 4 bit，下地址字段长 9 bit，从而可知控制字段长

$$(48 - 4 - 9) = 35 \text{ bit.}$$

(2) 微指令采用水平型格式，后继微指令地址采用断定方式，绘制微程序控制器逻辑框图如下。

