

一、

1. (1) 指令集架构层 (2) 微代码层 (3) 编译器 (4) 一一
2. (5) 数据位数 (6) 机器字长 (7) 字节对齐 (8) 小端
3. (9) 动作 (10) 平均时钟周期数 (11) IPC (12) $MIPS = 1/(t * CPI)$ (13) MFLOPS
4. (14) 存储程序 (15) 程序控制 (16) 存储器 (17) 控制器 (18) 总线
5. (19) 符号和数值 (20) 原码 (21) 符号位可以直接参与运算 (22) 移码 (23) 2^n
6. (24) 机器 0 (25) 1 (26) 11.0...或 00.1...
7. (27) 检错 (28) 纠错 (29) 检验位 (30) 码距 (31) 多重 (32) 4 (33) 2 (34) 1 (35) 整除
8. (36) 行波进位 (37) 先行 (38) 生成 (39) 传递
9. (40) 位扩展 (41) 顺序编址 (42) $T = m \tau$
10. (43) 随机 (44) 顺序 (45) 位置 (46) 直接
11. (47) 开关 (48) 熔丝 (49) 1 (50) 浮置栅 (51) 紫外线 (52) 某一 (53) 非易失
12. (54) 异步 (55) 列地址 (56) 列地址计数器 (57) 突发式 (58) 流水线 (59) 时钟 (60) 双倍

二、

1. DRAM 利用存储电容的电荷表示数据，电容充、放电的过程较慢，所以其读、写速度都比利用 MOS 导通截止的 SRAM 单元要慢。DRAM 的集成度更高。这是因为 DRAM 的存储单元设计相对简单，每个存储单元由一个电容和一个晶体管组成，这种设计使得 DRAM 能够在相同的芯片面积上容纳更多的存储单元。相比之下，SRAM 的存储单元由 6 个晶体管组成，结构更为复杂，因此其集成度相对较低。

2. 全相联映射：主存地址分为主存块地址 tag 和块内偏移 offset，主存中的每一个数据块都可以放置到 cache 的任意一个数据块中，是一对多的映射关系。主存块地址 tag 字段将同时与所有行中的标记字段 tag 进行多路并发比较，有多少行就需要设置多少个比较器。

直接相联映射：区地址 tag+行索引 index+块内偏移 offset。Cache 行号 $i = \text{主存块号 } j \bmod (\text{cache 行数 } n)$ 。只需要设置一个共享的比较器即可完成查找。

组相联映射：标记 tag+组索引 index+块内偏移 offset。Cache 组号 = 主存组号 \bmod (cache 组数)。Cache 内有多少组就需要多少比较器

3. 1) 符号为参与运算，所以 R0 寄存器的位宽为 $n+1$ ，乘数寄存器 R1 末端增设了附加位 y_{n+1} ，且 y_{n+1} 初值为 0，R1 位宽位 $n+2$

2) 每次通过移位，把低 $n+1$ 位存放在 R1 高位中，并把最低两位连接到 C_{in} 端

3) 每次对乘数寄存器中 y_n, y_{n+1} 两位进行判断，并根据 $y_{n+1}y_n$ 的值决定累加值，图中利用 $y_n y_{n+1}$ 控制多路选择器实现了 $+0, +[x]$ 补, $+[-x]$ 补的操作，减法操作是通过反码末位加一的方式实现的，图中利用加法器进位输入实现了末位加一

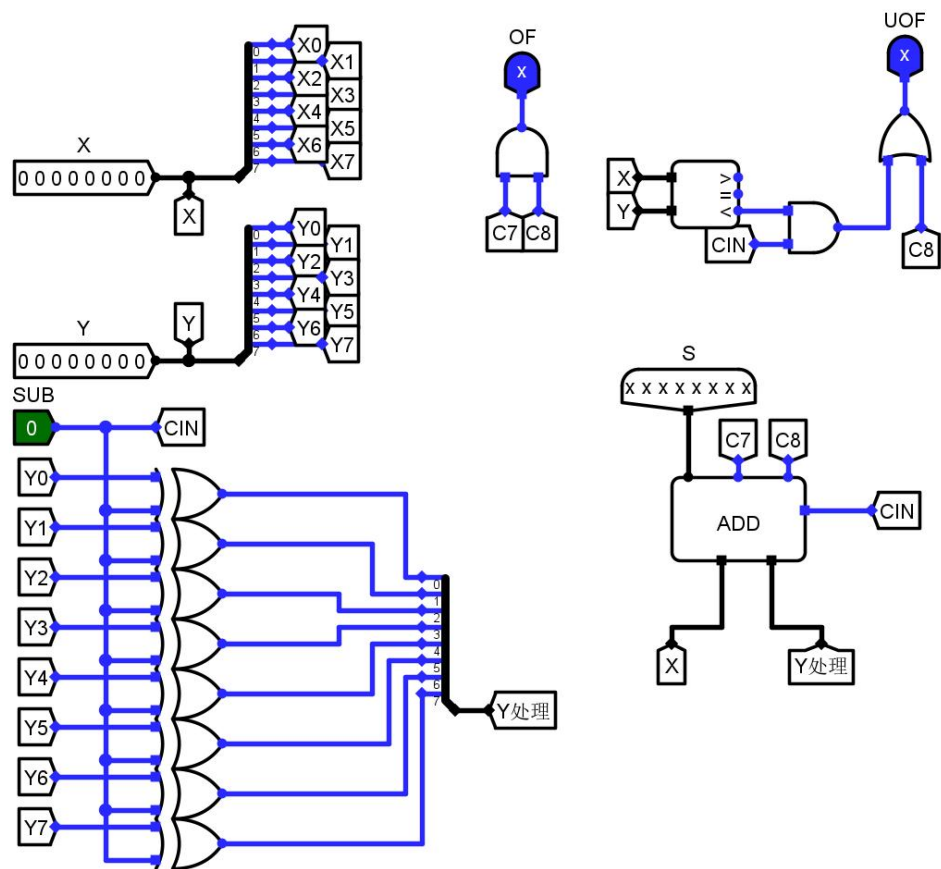
4) $\{P, y\} = \{(P + (y_{n+1} - y_n)[x] \text{ 补}), y\} / 2$

4. 1) TLB 本质上就是一个容量较小的 cache，TLB 离 CPU 更近，访问速度更快。页表则是存储在主存中，CPU 优先访问 TLB，只有当 TLB 未命中时，才回去主存中找页表，然后替换到 TLB 中

2) MMU 利用页表基址寄存器 PTBR 和虚页号生成页表项地址 PETA

3) 第 6 步，返回的 PTE 中有效位为 1，证明所访问的页表不在 TLB 中，而在主存之中，所以更新 TLB 表

4) 删除步骤 4,5



5.