



# 厦门大学《计算机组成原理》课程试卷

## 软件学院 软件工程系 2016 级 软件工程专业

主考教师：曾文华 张海英

试卷类型：(B 卷) 答案

### 一、填空题（共 28 分，每空 1 分）

1. 从计算机系统的多级层次结构来看，硬件研究主要涉及 M1 实际机器 和 M0 微程序机器 两部分。
2. 总线是各部件 共享 的传输介质，总线带宽就是总线的 数据传输率，目前的计算机普遍采用 多总线 结构。
3. 指令一般划分为 取指周期、间指周期、执行周期 和 中断周期 四个工作周期。
4. 对于 24 位地址线的主存而言，按字节寻址的范围是 16M(B)，按字寻址的范围则是 8M(B)。
5. 用海明码对长度为 8 位的数据进行检/纠错时，若能纠正一位错，则校验码位数至少为 4 位。检测位是按照 奇偶校验 原则进行检测的。
6. Cache 高速缓冲存储器是根据程序访问的 局部性能 原理设计的。为了体现这一基本原理，Cache 读操作，如果 不命中 且 Cache 满 时，需要进行页面替换。Cache 写操作时，必须与主存保持 一致。
7. 中断服务程序的入口地址是通过 向量 地址寻找的，该地址是由 硬件自动 产生的，单重中断与多重中断的区别在于 开中断 指令出现的位置。
8. 系统的并行性包含 同时性 和 并发性 两个方面的含义。
9. PROM 是可以实现 一 次编程的 只读 存储器，而 EPROM 可以实现 多 次的改写，是通过 紫外线 照射实现的。
10. 数据相关主要包括 RAW、WAR 和 WAW，采取的主要措施是后推法和 旁路法。

### 二、简答题（共 48 分，7 题）

1. 设  $X = -0.1101, Y = -0.1011$ , 求  $[XY]_{补}$  (5 分)

解：采用 Booth 算法（补码一位乘法）

$[X]_{\text{补}}=11.0011$        $[Y]_{\text{补}}=1.0101$        $[-X]_{\text{补}}=00.1101$

部分积	乘数	附加位	说明
	(一开始为 $[Y]_{\text{补}}=1.0101$ )	(一开始为 0)	
00.0000	1.010 <b>1</b> <b>0</b>		10 则 $+[-X]_{\text{补}}$
+ 00.1101			
00.1101	10101	0	右移 1 位
00.0110	1101 <b>0</b> <b>1</b>		01 则 $+ [X]_{\text{补}}$
+ 11.0011			
11.1001	11010	1	右移 1 位
11.1100	1110 <b>1</b> <b>0</b>		10 则 $+ [-X]_{\text{补}}$
+ 00.1101			
00.1001	11101	0	右移 1 位
00.0100	1111 <b>0</b> <b>1</b>		01 则 $+ [X]_{\text{补}}$
+ 11.0011			
11.0111	11110	1	右移 1 位
11.1011	1111 <b>1</b> <b>0</b>		10 则 $+ [-X]_{\text{补}}$
+ 00.1101			
00.1000	1111 <b>1</b>	0	

结果为： $[X*Y]_{\text{补}}=0.1001\ 1111$

2. 一个  $8K \times 8$  位的动态 RAM 芯片，其内部结构排列成  $256 \times 256$  形式，读/写周期为  $0.1\ \mu s$ 。试回答下述问题：（7 分）

(1) 动态随机存储器除了正常的读写，还需要额外的刷新工作，为什么？（2 分）

答：电容存在电荷泄漏的问题，使得“1”的内容被改写。

- (2) 何为“死时间”？采用集中式刷新、分散刷新和异步刷新的“死区”各是多少？应该如何安排消除“死区”？（5分）

答：内存无法进行正常的读写操作，为“死时间”。 1分

集中式刷新的死区时间是  $25.6 \mu s$  （ $256 \times 0.1 = 25.6$ ） 1分

分散刷新的死区时间是 0 1分

异步刷新的死区时间是  $0.1 \mu s$  1分

如果将刷新放在指令的译码阶段，则可以消除掉“死区” 1分

3. 有一 Cache 系统，字长为 16 位，主存容量为 16 字  $\times$  256 块，Cache 的容量为 16 字  $\times$  8 块。采用全相联地址映射方式，求：（8分）

- (1) 主存和 Cache 的字地址各为几个 bit？

答：Cache 8 块（3 位）、16 个字（4 位） 共 7 位 1分

主存 256 块（8 位）、16 个字（4 位） 共 12 位 1分

Cache 地址	3 位	4 位
主存地址	8 位	4 位

- (2) 如果原先已经依次装入 5 块的信息，问字地址为 338H 所在的主存块将装入 Cache 块中的哪一块以及在 cache 中的字地址？ 2分

答：主存地址 = 338H = 0011 0011 1000

既然是全相联，就是主存可以进入 Cache 任何一块的位置。

Cache 目前已经装入 5 块，故该主存块装入 Cache 的第 6 块（下标从 0 开始，所以实际对应下标为 101）。字地址为：101 1000 = 58H

- (3) 如果块表中地址为 1 的行中，标记着 36H 的主存块号标志，（且该主存标志对应的 Cache 块号为 101B），则当 CPU 送来主存的字地址为 368H 时，是否命中（1分）？如果命中，此时的 cache

的字地址为多少？（2 分）

答：主存地址 368H，对应的高位（主存字块标志）是 36H，根据块表，其命中。

对应的 Cache 地址为 101 1000

（101 为 Cache 块号， 1000 为 368H 中的 8）

4. 已知程序查询方式的接口电路如图 1 所示，请回答下述问题：（6 分）

1) 触发器 D 和 B 的作用是什么？（3 分）

答：D 和 B 反映外设的工作状态触发器（D 是完成触发器，B 是工作触发器）

- D=0, B=0, 表示 I/O 设备处于暂停状态
- D=1, B=0, 表示 I/O 设备处于准备就绪状态
- D=0, B=1, 表示 I/O 设备处于工作状态

2) 为何要设置数据缓冲寄存器？（1.5 分）

答：解决外设与 CPU 的速度差

3) 准备就绪状态是由哪个信号触发的？（1.5 分）

答：由 D 触发器触发

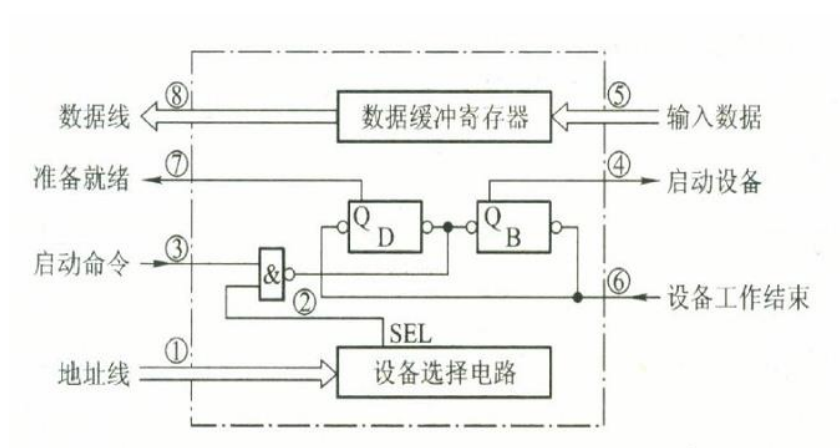


图 1 程序查询方式的接口电路

5. 图 2 为在 DMA 的工作方式中，CPU 暂停方式的时间示意图，请回答下述问题：（7 分）

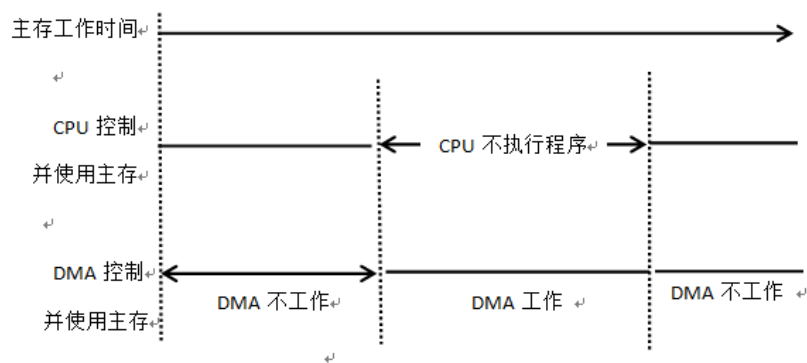


图 2 停止 CPU 访存

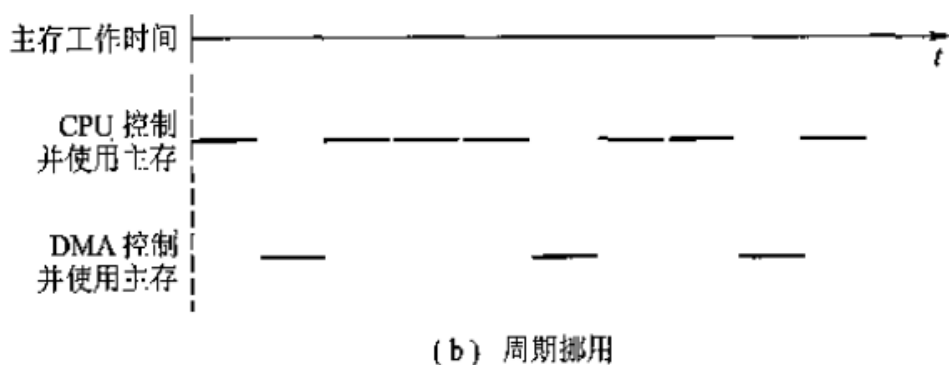
(1) 该方式的优缺点是什么？（2 分）

答：优点：控制简单，适用于数据传输率很高的 I/O 设备。

缺点：CPU 基本处于不工作状态或维持原有状态，对主存的利用率没有得到充分的发挥。

(2) 如果采用周期挪用方式，则有何不同？请参照图 2 绘出时间示意图（5 分）

答：周期挪用中，每当 I/O 设备发出 DMA 请求时，I/O 设备便挪用或窃取总线占用权一个或几个主存周期；而 DMA 不请求时，CPU 仍继续访问主存。周期挪用方式的时间示意图如下：



6. 设某机器有三个中断源 1、2、3，其优先级按  $1 > 2 > 3$  降序排列。假设中断处理时间均为  $t$ ，在图 3 所示的时间内共发生 5 次中断请求，图中①表示 1 级中断源发出中断请求信号，其余类推。（8 分）

(1) 画出 CPU 执行程序的轨迹。（5 分）

(2) 如果优先级改为  $2 > 1 > 3$ ，请给出对应的屏蔽字。（3 分）

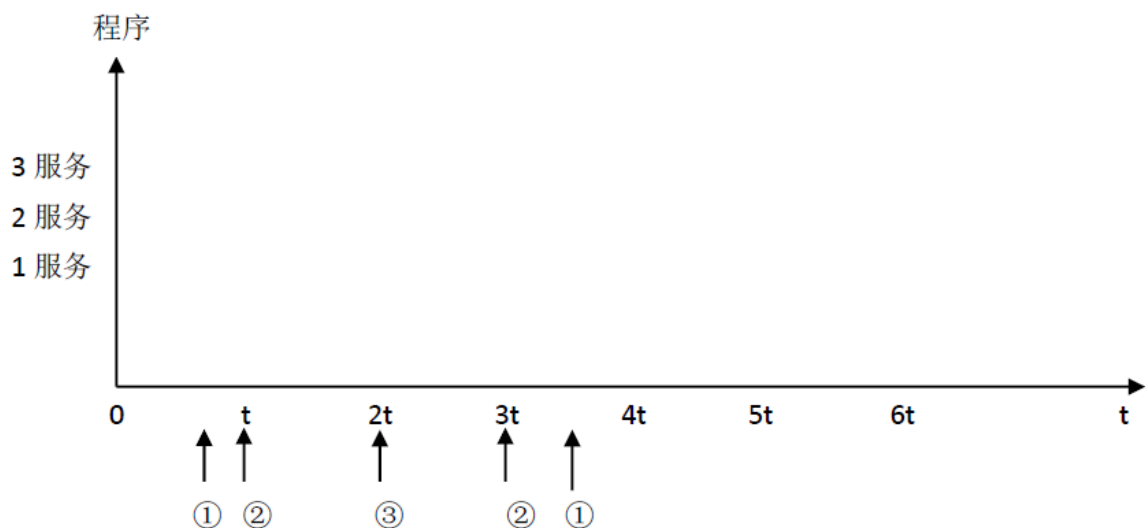
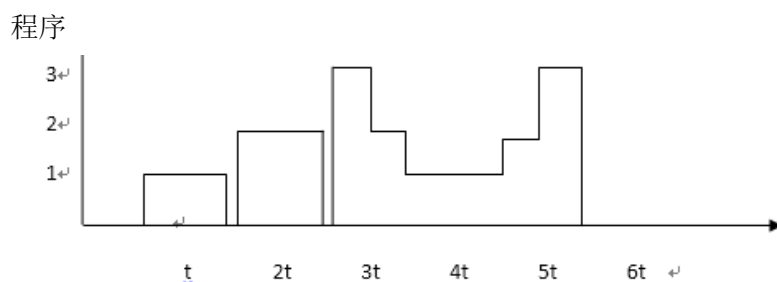


图 3 设置中断请求的时刻

答：（1）CPU 执行程序的轨迹如下所示：



(2)屏蔽字 3"

优先级	2	1	3	
2	1	1	1	
1	0	1	1	
3	0	0	1	

7. 某主存储器部分单元的地址码与存储器内容对应关系如表 1 所示，请回答下述问题：（7 分）

(1) 若采用寄存器间址方式读取操作数，指定寄存器 R0 的内容为 1000H，则操作数是多少？

答： A307H (R0)=(1000H)=A307H

(2) 若采用自减型寄存器间址方式  $-(R1)$  读取操作数,R1 内容为 1003H，则操作数是多少?指令执行完后 R1 的内容是多少？

答： F03CH R1=1002H (R1)=(1003H)=F03CH R1-1=1003H-1=1002H -> R1

(3) 若采用变址寻址方式  $X(R2)$ 读取操作数，指令中给出形式地址  $d=3H$ ，变址寄存器 R2 内容为 1000H，

则操作数是多少？

答：F03CH      有效地址= $R2+d=1000H+3=1003H$       (1003H)=F03CH

(4) 采用寄存器间址方式，指定寄存器  $R3=1002H$ ，则间址两次@ (R3)是多少？

答：D024H      ((R3))=((1002H))=(1004H)=D024H

表 1 地址码与存储内容的对应关系

地址码	存储内容
1000H	A307H
1001H	0B3FH
1002H	1004H
1003H	F03CH
1004H	D024H

### 三、设计题

1. 设CPU内部采用总线连接方式，如图4所示，完成下述任务：（13分）

(1) 写出完成STAX(X 为主存地址) 指令的全部微操作和控制信号。

(2) 写出ADD @X(X 为主存地址)指令的全部微操作和控制信号。

答：（1）指令 STAX 的取指周期的全部微操作和控制信号：

微操作

控制信号

PC → Bus → MAR

$PC_o$      $MAR_i$

1 → R

存储器读

M(MAR) → MDR

MDR → Bus → IR

$MDR_o$      $IR_i$

OP(IR) → CU

(PC) + 1 → PC

指令 STA X 的执行周期的全部微操作和控制信号：

微操作

控制信号

MDR → Bus → MAR                      MDR<sub>0</sub>    MAR<sub>i</sub>

AC → Bus → MDR                      AC<sub>0</sub>    MDR<sub>i</sub>

1 → W                      存储器写

MDR → M(MAR)

(2) 指令 ADD @X 的取指周期的全部微操作和控制信号：

微操作

控制信号

PC → Bus → MAR                      PC<sub>0</sub>    MAR<sub>i</sub>

1 → R                      存储器读

M(MAR) → MDR

MDR → Bus → IR                      MDR<sub>0</sub>    IR<sub>i</sub>

OP(IR) → CU

(PC) + 1 → PC

指令 ADD @X 的间指周期的全部微操作和控制信号

微操作

控制信号

MDR → Bus → MAR                      MDR<sub>0</sub>    MAR<sub>i</sub>



1 -> R 存储器读

## 存储器读

M(MAR) -> MDR

指令 ADD @X 的执行周期的全部微操作和控制信号

## 微操作

控制信号

MDR  $\rightarrow$  Bus  $\rightarrow$  MAR                      MDR<sub>0</sub>    MAR<sub>i</sub>

$$\text{MDR}_o \quad \text{MAR}_i$$

1->R	存储器读
------	------

## 存储器读

M(MAR)->MDR

MDR → Bus → Y                      MDR<sub>0</sub>    Y<sub>i</sub>

MDRo Yi

(AC) + Y -> AC	AC <sub>0</sub>	ALU <sub>i</sub>	+ (ALU 加法)
----------------	-----------------	------------------	------------

$$AC_0 \quad ALU_i \quad + (ALU \text{ 加法})$$
$$ALU_i + (ALU \text{ 加法})$$

+ (ALU 加法)

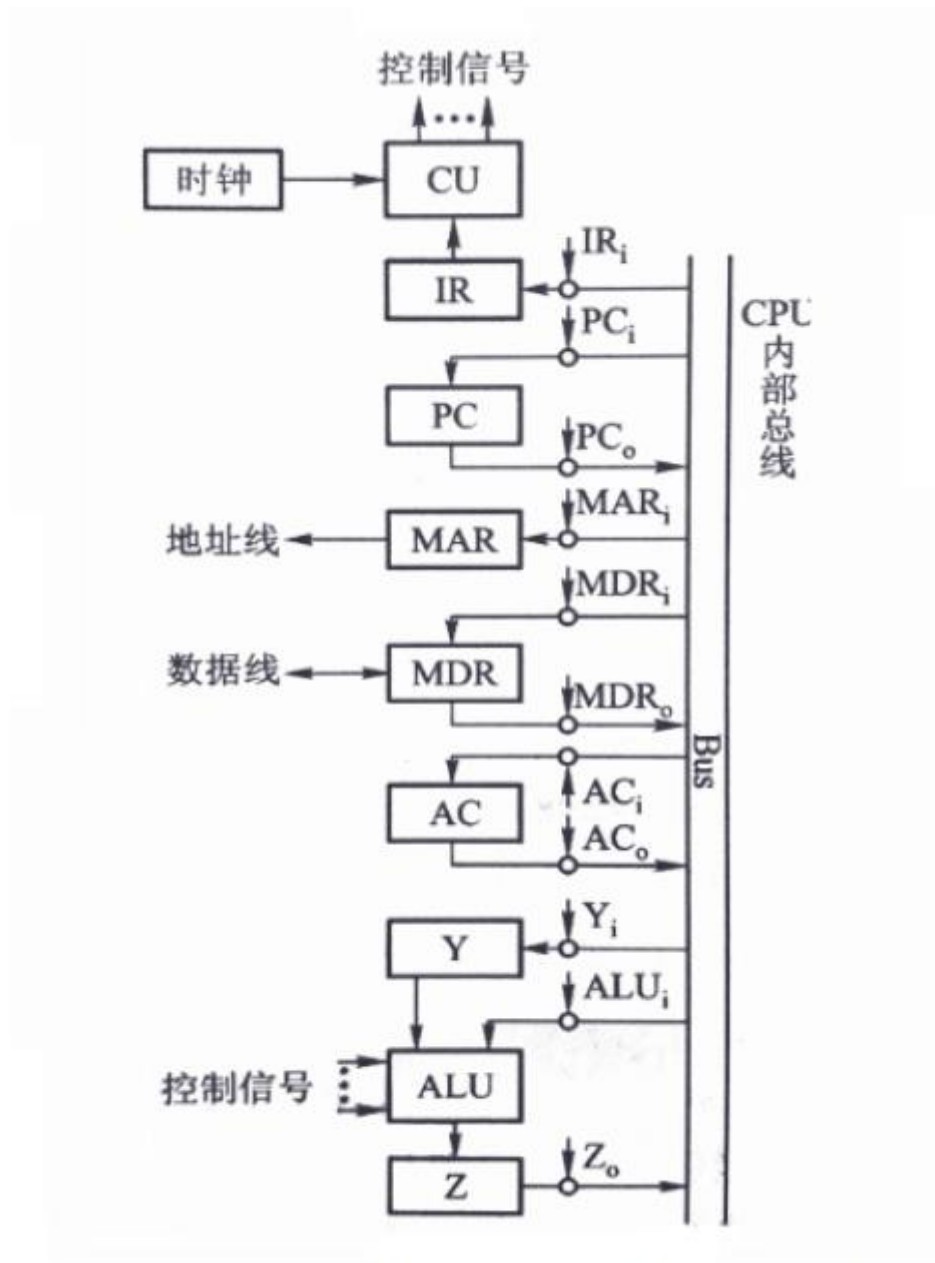


图 4 CPU 内部总线的数据通路和控制信号

2. 设CPU共有16根地址线，8根数据线，并用-MREQ作为访存控制信号（低电平有效），用-WR作为读写控制信号（高电平为读，低电平为写）。现有如下规格的存储芯片：1K×4位RAM，4K×8位RAM，2K×8位ROM，以及74138译码器和各种门电路，如图5所示。画出CPU与存储器连接图，要求：

（11分）

- (1) 主存地址空间分配：最大2K为系统程序区；相邻2K为用户程序区。
- (2) 合理选用上述芯片，说明各选几片？
- (3) 详细画出存储芯片的片选逻辑。

答：

(1) 最大 2K: F800H - FFFFH

1111 1000 0000 0000

1111 1111 1111 1111

相邻 2K: F000H - F3FFH (1K)

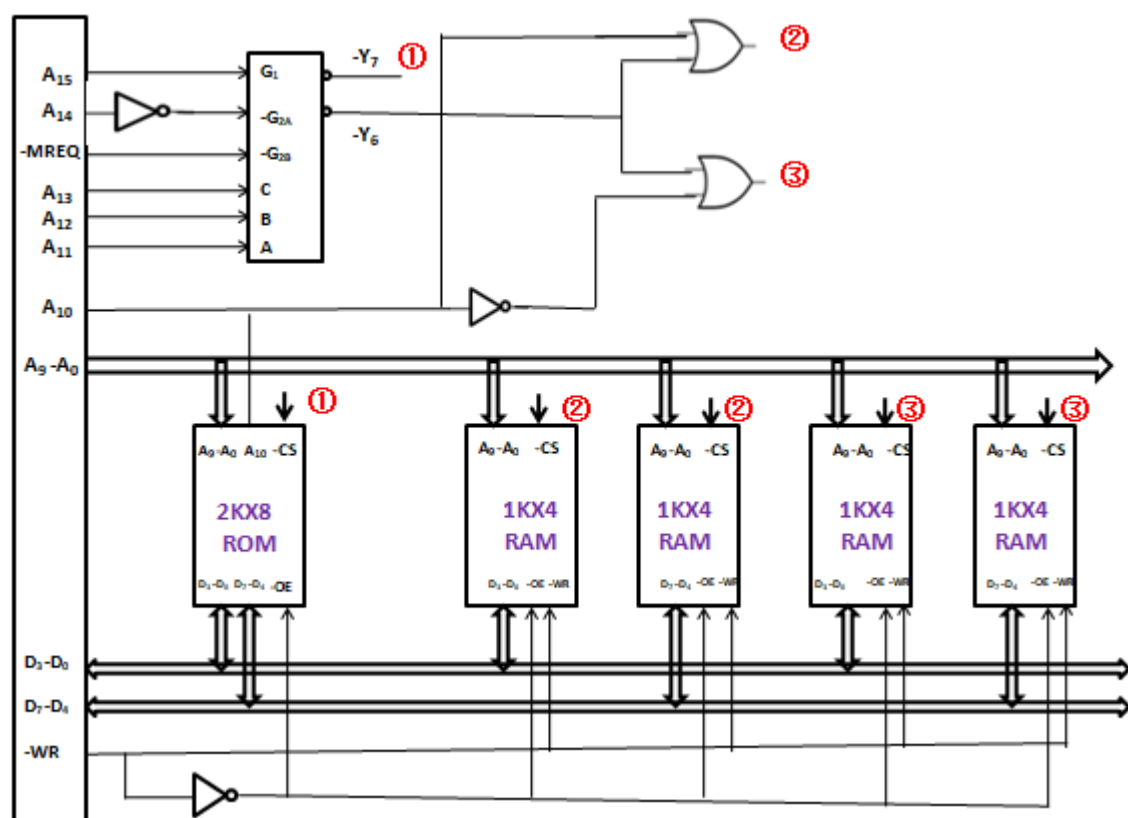
1111 0000 0000 0000

1111 0011 1111 1111

F400H - F7FFH (1K)

1111 0100 0000 0000

1111 0111 1111 1111



(1) 最大 2K 选用 1 片  $2K \times 8$  位的 ROM，相邻 2K 选用 4 片  $1K \times 4$  位的 RAM<sup>+</sup>

(2) 片选逻辑<sup>+</sup>

<sup>+</sup>

