



# 简单cpu的设计

智能1601

樊龙

201608010325



# 目录

## CONTENTS

ENTER YOUR COMPANY NAME



cpu设计步骤



cpu规格



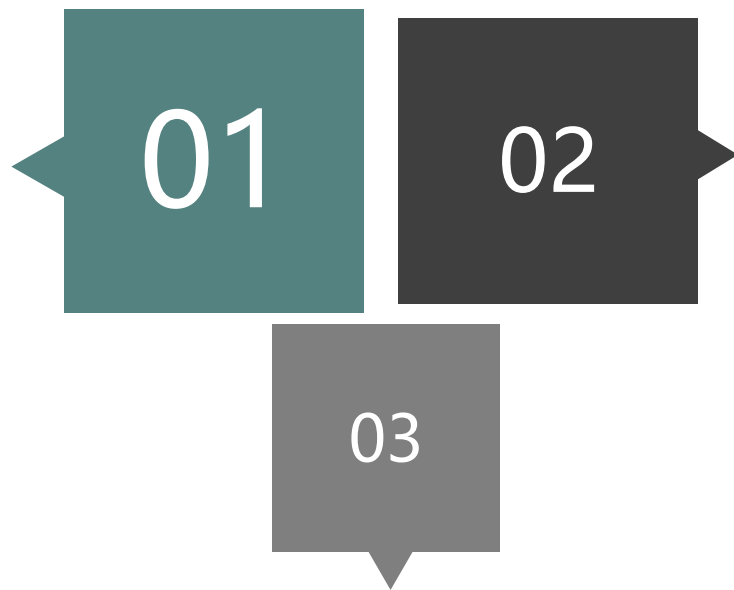
指令集



寄存器

## 01：确定它的用途

关键：使CPU的处理能力和它所执行的任务匹配。



## 02：设计指令集结构

## 03：设计状态图

- 列出在每个状态中执行的微操作
- 从一个状态转移到另外一个状态的条件

## CPU执行如下的操作序列

取指令周期：从存储器中  
取出一条指令，然后转到  
译码周期。

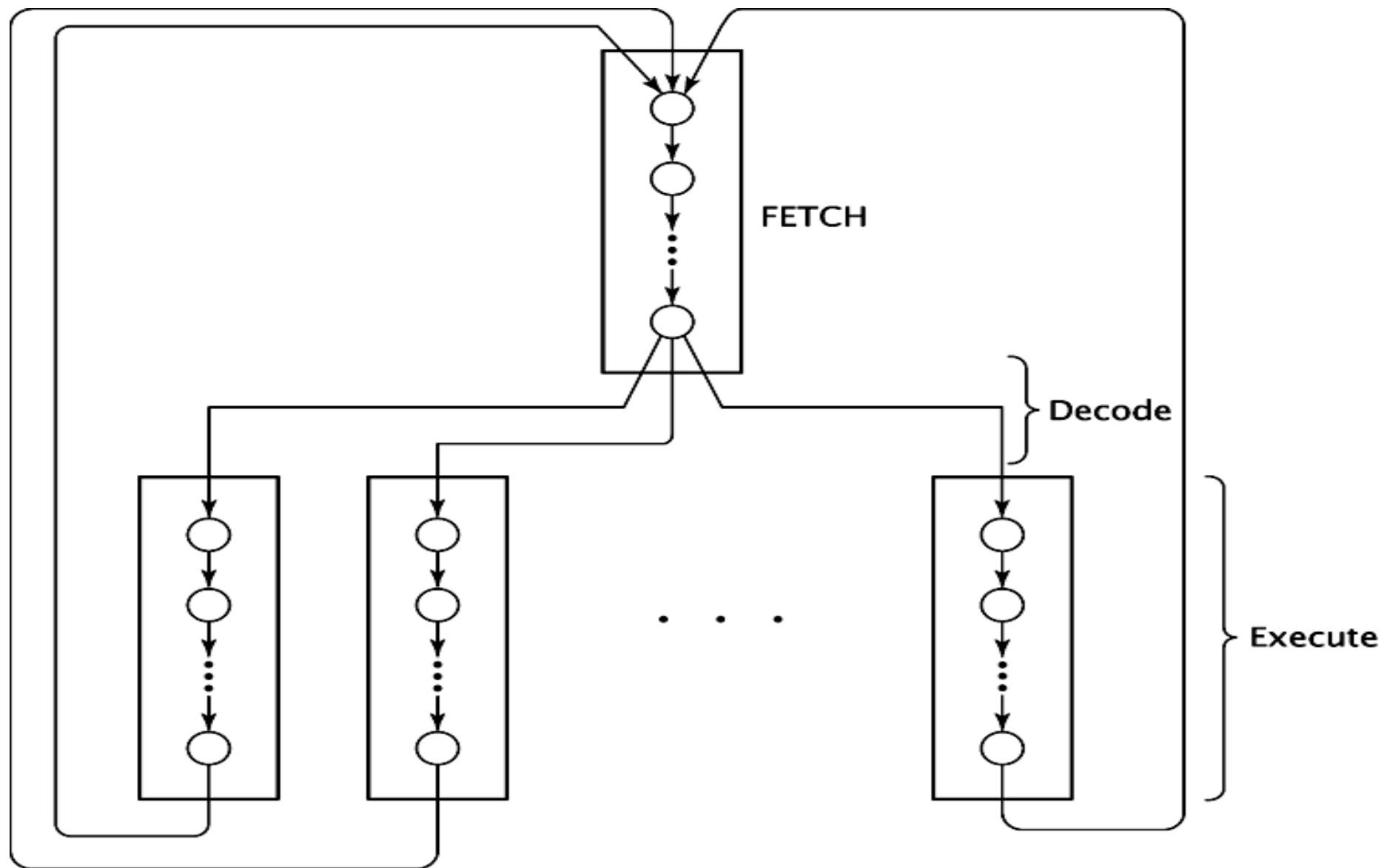
执行周期：执行该指令，  
然后转移到取指令周期去  
取下一条指令



译码周期：对该指  
令进行译码，即确  
定取到的是哪一种  
指令，然后转移到这  
种指令对应的执行周  
期。

# 状态图

一般cpu状态图:



## cpu的规格

1. 64K字节的存储器，每个存储单元8位宽。

地址引脚A[15..0]

数据引脚D[7..0]

2. CPU的三个内部寄存器

- ◆ 8位累加器AC：接受任何算术或者逻辑运算的结果，并为使用两个操作数的算术或者逻辑操作指令提供一个操作数。
- ◆ 寄存器R：一个8位通用寄存器。它为所有的双操作数算术和逻辑运算指令提供第二个操作数。它也可以用来暂时存放累加器马上要用到的数据。（减少存储器访问次数提高CPU的性能）
- ◆ 零标志位Z：每次执行算术运算或者逻辑运算的时候，它都将被置位。

# 指令集

指令	指令码	操作
NOP	0000 0000	无
LDAC	0000 0001 Γ	$AC \leftarrow M[\Gamma]$
STAC	0000 0010 Γ	$M[\Gamma] \leftarrow AC$
MVAC	0000 0011	$R \leftarrow AC$
MOV R	0000 0100	$AC \leftarrow R$
JUMP	0000 0101 Γ	GOTO Γ
JMPZ	0000 0110 Γ	IF (Z = 1) THEN GOTO Γ
JPNZ	0000 0111 Γ	IF (Z = 0) THEN GOTO Γ

# 指令集

ADD	0000 1000	$AC \leftarrow AC + R$ , IF $(AC + R = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
SUB	0000 1001	$AC \leftarrow AC - R$ , IF $(AC - R = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
INAC	0000 1010	$AC \leftarrow AC + 1$ , IF $(AC + 1 = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
CLAC	0000 1011	$AC \leftarrow 0$ , $Z \leftarrow 1$
AND	0000 1100	$AC \leftarrow AC \wedge R$ , IF $(AC \wedge R = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
OR	0000 1101	$AC \leftarrow AC \vee R$ , IF $(AC \vee R = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
XOR	0000 1110	$AC \leftarrow AC \oplus R$ , IF $(AC \oplus R = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$
NOT	0000 1111	$AC \leftarrow AC'$ , IF $(AC' = 0)$ THEN $Z \leftarrow 1$ ELSE $Z \leftarrow 0$



有那些寄存器？寄存器的功能是什么？

- ◆ 16位的地址寄存器AR：通过引脚A[15..0]向存储器提供地址。
- ◆ 16位的程序计数器PC：存放的是将要执行的下一条指令的地址，或者指令需要的下一个操作数的地址。

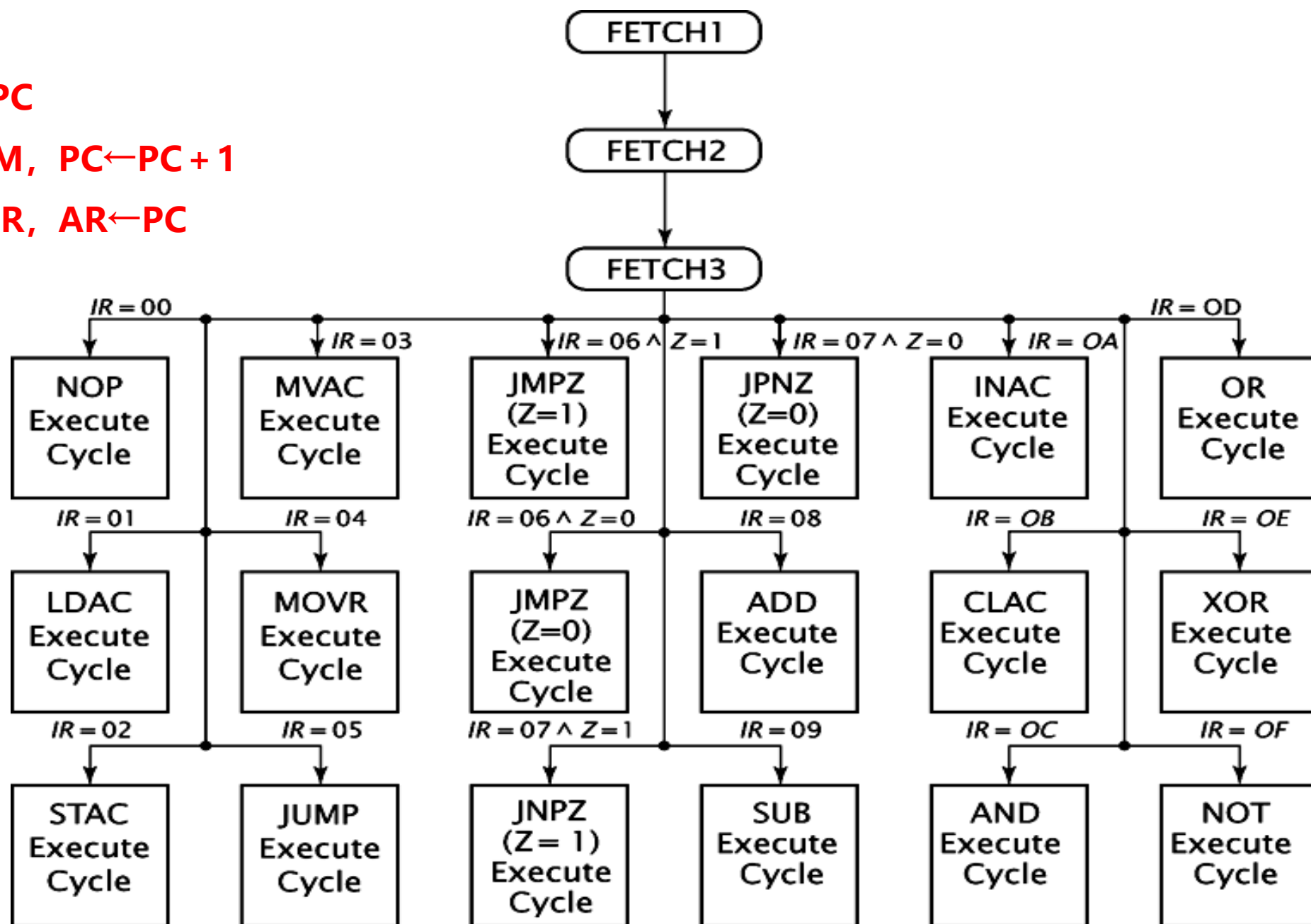
AR和PC必须能够执行并行的装载和递增的操作。两个寄存器都从内部总线上接受数据。

- ◆ 8位的数据寄存器DR：通过D[7..0]从存储器中接收指令和数据并且向存储器传送数据。
- ◆ 8位的指令寄存器IR：存放的是从存储器中取出来的操作码。
- ◆ 8位的临时寄存器TR：在指令执行过程中，临时存储数据。（程序员不能访问）
- ◆ DR, IR, R, TR必须能够并行装载数据。  
CPU用一个ALU来完成所有这些功能。  
ALU能够接受AC的数据作为一个输入，接受内部总线上的数据作为另外一个输入。AC总是从ALU得到它的输入。CPU同时也根据ALU的输出来决定结果是否0，从而设置Z。

**FETCH1:  $AR \leftarrow PC$**

**FETCH2:  $DR \leftarrow M, PC \leftarrow PC + 1$**

**FETCH3:  $IR \leftarrow DR, AR \leftarrow PC$**



# 数据通路

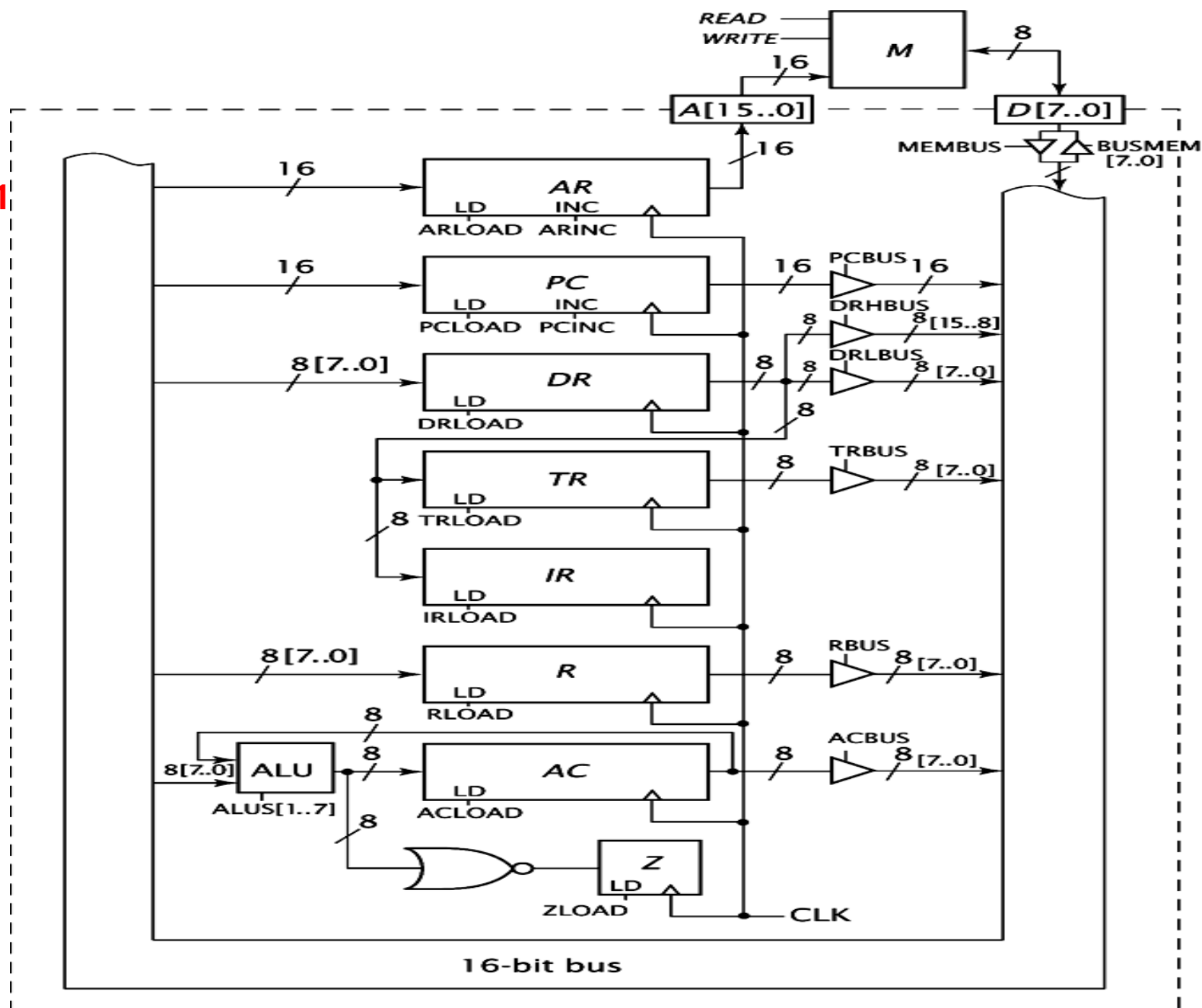
STAC1:  $DR \leftarrow M$ ,  $PC \leftarrow PC + 1$ ,  $AR \leftarrow AR + 1$

STAC2:  $TR \leftarrow DR$ ,  $DR \leftarrow M$ ,  $PC \leftarrow PC + 1$

STAC3:  $AR \leftarrow DR$ ,  $TR$

STAC4:  $DR \leftarrow AC$

STAC5:  $M \leftarrow DR$





感谢您的观看

THANK YOU FOR YOUR WATCHING

PLACED LOGO