

INTEL

微型计算机系列器件手册

中 册

上海交通大学科技交流室
微机研究室

一九八三年十二月

第七章 MCS-80/85™ 系列微计算机

8080A/8080A-1/8080A-2

8 位 N 沟道微处理器

- 与 TTL 电路直接兼容
- 指令周期为 2 微秒(8080A-1 为 1.3 微秒 8080A-2 为 1.5 微秒)
- 有较强功能的指令系统。
- 6 个通用寄存器和一个累加器。
- 16 位程序计数器对内存 64K 字节有直接寻址能力。
- 16 位堆栈指示器和堆栈, 可程序快速变换的操作指令
- 十进制, 二进制和双倍精度运算。
- 能提供优先级的向量中断。
- 有 512 个直接寻址的输入/输出转接口。

英特尔公司的 8080A 是一种完备的 8 位并行中央处理器(CPU), 采用 N 沟道硅栅 MOS 工艺组合在单片大规模集成电路上, 这就为用户提供一个高性能解决控制和处理应用的手段。

8080A CPU 中含有 6 个 8 位通用工作寄存器和一个累加器, 6 个通用寄存器可以单个地或成对地寻址, 以提供单精度及双倍精度的操作数, 算术和逻辑指令可使 4 个条件标志置位或复位, 第 5 个标志供十进制算术运算使用。

8080A 的堆栈设置在外部, 使得存贮器的任何区域都可作为后进先出的堆栈, 该堆栈用于存放或恢复累加器, 标志寄存器, 程序计数器以及 6 个通用寄存器的内容。16 位堆栈指示器控制外部堆栈的寻址, 这个堆栈提供 8080A 易于处理多级优先级中断, 并能迅速地存放和恢复处理器的状态, 还能提供几乎是无限次的子程序嵌套。

该微处理器设计得能简化系统设计, 把 16 根地址线和 8 根双向数据总线分开, 使得处理器易与存贮器及输入/输出接口连接, 存贮器及输入/输出接口控制信号由 8080A 直接提供, 出现 HOLD 信号时对地址和数据总线作特殊的控制, 可把处理器操作挂起, 并强制地址及数据总线进入高阻抗状态, 能使这些总线和其他控制设备进行“或”的操作, 保证直接存取存贮器或进行多处理器操作。

注意: 8080A 功能及电性能同英特尔 8080 兼容。

8080A 功能框图见图 1 所示, 引脚排列见图 2 所示。

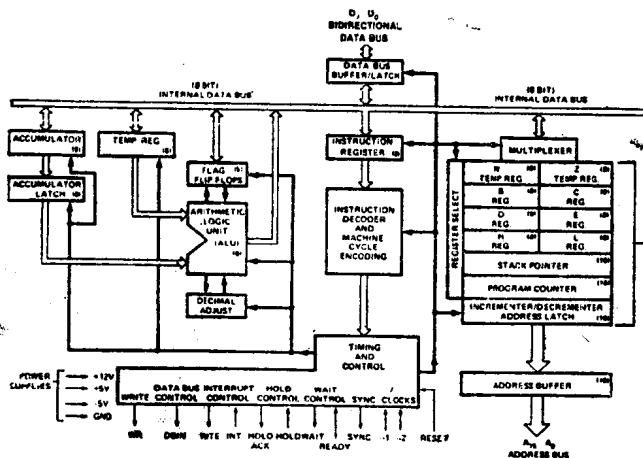


图 1 8080A 功能框图

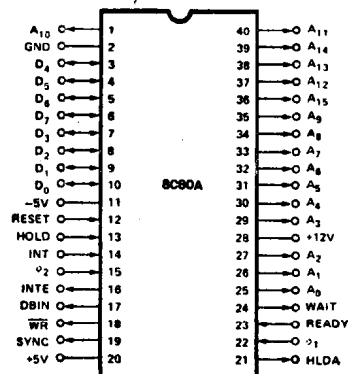


图 2 引脚排列图

表 1 引脚说明

符 号	类 型	名 称 及 功 能
A ₁₅ - A ₀	O	Address Bus(地址总线): 地址总线提供高达 64K 的 8 位字的存储器地址或者表示 256 个输入/输出的 I/O 设备, A ₀ 是最低有效地址位。
D ₇ - D ₀	I/O	Data Bus(数据总线): 数据总线为 CPU 和存储器或 I/O 设备之间进行指令和数据传送提供双向通讯, 在每个机器周期的第一时钟周期, 8080A 输出一个状态字送数据总线, 指出现行的机器周期, D ₀ 为最低有效数据位。
SYNC	O	Synchronizing Signal(同步信号), 同步信号引出端用作指示每个机器周期开始的信号。
DBIN	O	Data Bus in(数据总线允许输入): 数据总线允许输入信号对外部电路指出数据总线处于输入方式。该信号用来开启数据从存储器或 I/O 至 8080A 数据总线的门控电路。
READY	I	Ready(准备就绪): 准备就绪信号表示在 8080A 数据总线上从存储器或输入的数据是有效的, 该信号作为 CPU 与慢速存储器或 I/O 设备同步工作。如果 8080A 送出一个地址后而没有接到 READY 信号输入, 8080A 根据 READY 信号是低电平, 就进入“等待”(WAIT)状态, READY 也能用于 CPU 单步工作。
WAIT	O	Wait(等待): 等待信号用来承认 CPU 处于等待(WAIT)状态。
WR	O	Write(写): 写信号用来控制存储器输入或 I/O 设备输出。当 WR 信号为有效低电平(WR=0)时, 数据总线上数据正是稳定的。
HOLD	I	Hold(保持请求): Hold 信号请求 CPU 进入 HOLD 状态。HOLD 状态允许外设紧接着 8080A 完成现行机器周期对地址和数据总线的使用后, 取得对这些总线的控制。它是在下述条件下被 8080A 识别的, • CPU 处于 HALT(暂停)状态中。 • CPU 处于 T₂ 或 T_W 状态中, 而 READY 信号已有效进入 HOLD 状态的结果 CPU 的地址总线(A₁₅ - A₀)及数据总线(D₇ - D₀)即进入高阻抗状态, CPU 用它的保持响应(HLDA), 引出端表示它已响应 HOLD 状态。
HLDA	O	Hold Acknowledge(保持响应): HLDA 信号出现在于响应 HOLD 信号, 并指出数据和地址总线即将转成高阻抗状态。HLDA 信号开始于: • 读存储器周期或输入周期的 T₃。 • 写存储器周期或输出操作时 T₃ 过后的时钟周期。 不论是那种情况, HLDA 信号都出现在 φ ₂ 的上升沿之后。

INTE	0	Interrupt Enable (中断允许): 指示内部允许中断触发器的内容, 这个触发器可由允许中断及禁止中断指令使之置位或复位, 当它复位时, 它禁止CPU接收中断。当接收一个中断后, 在指令取操作码周期(M ₁)的T ₁ 时间能自动复位(再禁止中断), 它也能用RESET信号来复位。
INT	1	Interrupt Request(中断): 在现行指令结束时, 或者是暂停(Halt)时CPU就用此信号来识别中断请求, 如果CPU处于保持请求(HOLD)状态或者中断允许触发器复位时, 那就不响应这个请求。
RESET	1	Reset(复位): 当RESET信号有效时, 程序计数器的内容被清除, Reset(复位)信号之后, 程序计数器将从零单元开始。允许中断(INTE)触发器和保持响应(HLDA)触发器也被复位。标志寄存器, 累加器, 堆栈指示器及寄存器都不清除。
V _{SS}		参考地
V _{DD}		Power(电源): +12±5% 伏
V _{CC}		Power(电源): +5±5% 伏
V _{BB}		电源Power(电源): -5±5% 伏
φ ₁ φ ₂		Clock Phases: 相时钟外部提供的两相时钟(不能与TTL电路相容的)

最大极限值

加偏压下的温度

0℃至+70℃

存放温度

-65℃至+150℃

对应 V_{BB} 所有输入或输出电压

-0.3V至+20V

对应 V_{BB}, V_{CC}, V_{DD} 及 V_{SS} 的电压

-0.3V至+20V

功率耗境

1.5W

备注: 高于表中引出的“最大极限值”条件会造成器件彻底损坏。这仅是一个恶劣的极限值, 器件在恶劣条件下, 或在下列规定所指出的操作条件之外的状态工作, 将不能保证它的操作功能, 在最大极限值条件下工作若干时间, 就可影响器件的可靠性。

直流特性

(T_A=0℃至70℃, V_{DD}=+12V±5%, V_{CC}=+5V±5%, V_{BB}=-5V±5%, V_{SS}=0V; 另有注释除外)

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Condition
V _{ILC}	Clock Input Low Voltage	V _{SS} -1		V _{SS} +0.8	V	$I_{OL}=1.9\text{mA}$ on all outputs, $I_{OH}=-150\mu\text{A}$.
V _{IHC}	Clock Input High Voltage	9.0		V _{DD} +1	V	
V _{IL}	Input Low Voltage	V _{SS} -1		V _{SS} +0.8	V	
V _{IH}	Input High Voltage	3.3		V _{CC} +1	V	
V _{OL}	Output Low Voltage			0.45	V	
V _{OH}	Output High Voltage	3.7			V	
I _{DD} (AV)	Avg. Power Supply Current(V _{DD})		40	70	mA	Operation T _{cy} =48μsec
I _{CC} (AV)	Avg. Power Supply Current(V _{CC})		60	80	mA	
I _{BB} (AV)	Avg. Power Supply Current(V _{BB})		.01	1	mA	

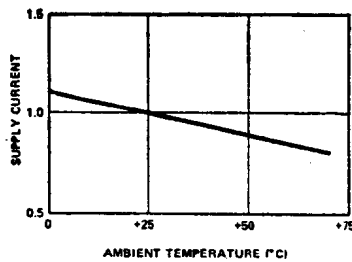
续 表

I_{IL}	Input Leakage			+ 10	μA	$V_{SS} \leq V_{IN} \leq V_{CC}$
I_{CL}	Clock Leakage			± 10	μA	$V_{SS} \leq V_{CLOCK} \leq V_{DD}$
I_{DL} [2]	Data Bus Leakage in Input Mode			- 100 - 2.0	μA mA	$V_{SS} \leq V_{IN} \leq V_{SS} + 0.8V$ $V_{SS} + 0.8V \leq V_{IN} \leq V_{CC}$
I_{FL}	Address and Data Bus Leakage During HOLD			+ 10 - 100	μA	$V_{ADDR/DATA} = V_{CC}$ $V_{ADDR/DATA} = V_{SS} + 0.45V$

电容

(T_A = 25℃, V_{CC} = V_{DD} = V_{SS} = 0V, V_{BB} = -5V)

Symbol	Parameter	Typ.	Max.	Unit	Test Condition
C _φ	Clock Capacitance	17	25	pf	f _c = 1MHz
C _{IN}	Input Capacitance	6	10	pf	Unmeasured Pins
C _{OUT}	Output Capacitance	10	20	pf	Returned to V _{SS}



典型的电源电流与温度关系(标准的)

注意: 1. RESET 信号至少应该在 3 个时钟周期是有效的。

2. 电源 $\Delta I / \Delta T_A = -0.45\% / ^\circ C$

交流特性(8080A)

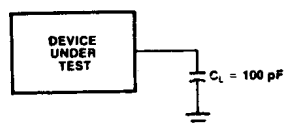
(T_A = 0℃ 至 70℃, V_{DD} = +12V ± 5%, V_{CC} = +5V ± 5%, V_{BB} = -5V ± 5%, V_{SS} = 0V, 另有注释除外)

Symbol	Parameter	Min.	Max.	-1 Min.	-1 Max.	-2 Min.	-2 Max.	Unit	Test Condition
t _{cy} [3]	Clock Period	0.48	2.0	0.32	2.0	0.38	2.0	μsec	
t _r , t _f	Clock Rise and Fall Time	0	50	0	25	0	50	nsec	
t _{φ1}	φ ₁ Pulse Width	60		50		60		nsec	
t _{φ2}	φ ₂ Pulse Width	220		145		175		nsec	
t _{D1}	Delay φ ₁ to φ ₁	0		0		0		nsec	
t _{D2}	Delay φ ₂ to φ ₁	70		60		70		nsec	
t _{D3}	Delay φ ₁ to φ ₂ Leading Edges	80		60		70		nsec	
t _{DA}	Address Output Delay From φ ₁		200		150		175	nsec	} C _L = 100 pF

续 表

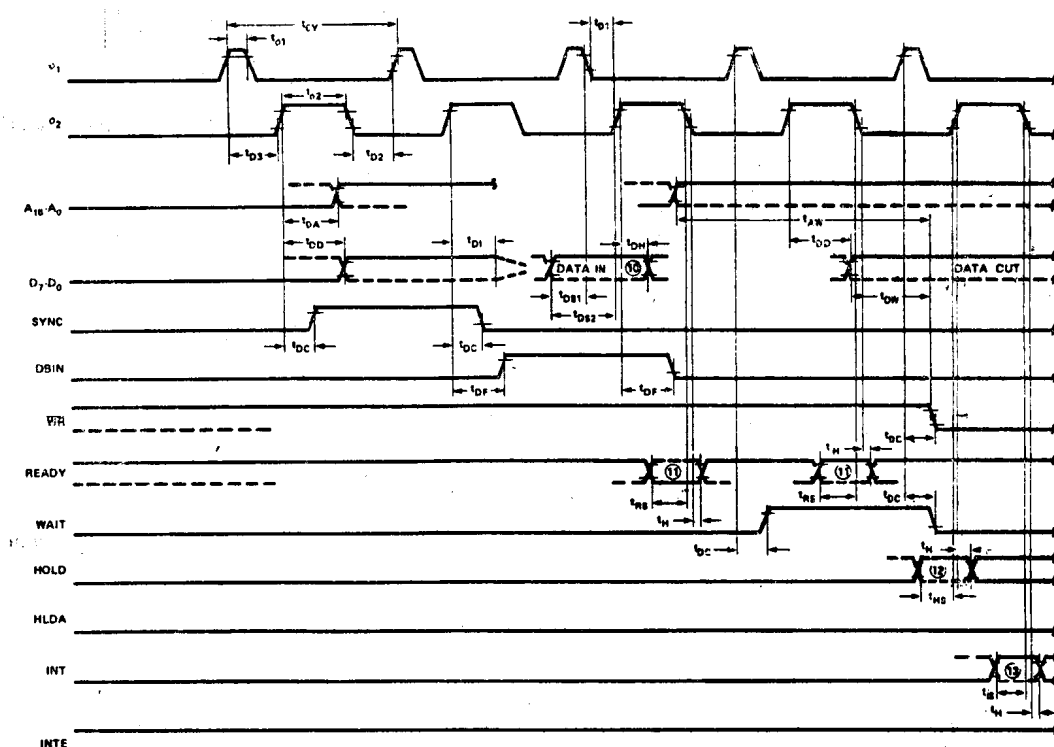
t _{DD}	Data Output Delay From ϕ_2		220		180		200	nsec	C _L = 50 pF
t _{DC}	Signal Output Delay From ϕ_2 or ϕ_2 (SYNC, WR, WAIT, HLDA)		120		110		120	nsec	
t _{DF}	DBIN Delay From ϕ_2	25	140	25	130	25	140	nsec	
t _{D1[1]}	Delay for Input Bus to Enter Input Mode		t _{DF}		t _{DF}		t _{DF}	nsec	
t _{DS1}	Data Setup Time During ϕ_1 and DBIN	30		10		20		nsec	C _L = 50 pF
t _{DS2}	Data Setup Time to ϕ_2 During DBIN	150		120		130		nsec	
t _{DH[1]}	Data Hold time From ϕ_2 During DBIN	[1]		[1]		[1]		nsec	
t _{IE}	INTE Output Delay From ϕ_2		200		200			nsec	
t _{RS}	READY Setup Time During ϕ_2	120		90		90	200	nsec	C _L = 50 pF
t _{HS}	HOLD Setup Time to ϕ_2	140		120		120		nsec	
t _{IS}	INT Setup Time During ϕ_2	120		100		100		nsec	
t _H	Hold Time From ϕ_2 (READY, INT, HOLD)	0		0		0		nsec	
t _{FD}	Delay to Float During Hold (Address and Data Bus)		120		120		120	nsec	C _L = 100 pF; Address, Data C _L = 50pF, WR, HLDA, DBIN
t _{AW}	Address Stable Prior to WR	[5]		[5]		[5]		nsec	
t _{DW}	Output Data Stable Prior to WR	[6]		[6]		[6]		nsec	
t _{WD}	Output Data Stable From WR	[7]		[7]		[7]		nsec	
t _{WA}	Address Stable From WR	[7]		[7]		[7]		nsec	
t _{HF}	HLDA to Float Delay	[8]		[8]		[8]		nsec	
t _{WF}	WR to Float Delay	[9]		[9]		[9]		nsec	
t _{AH}	Address Hold Time After DBIN During HLDA	-20		-20		-20		nsec	

交流负载测试电路



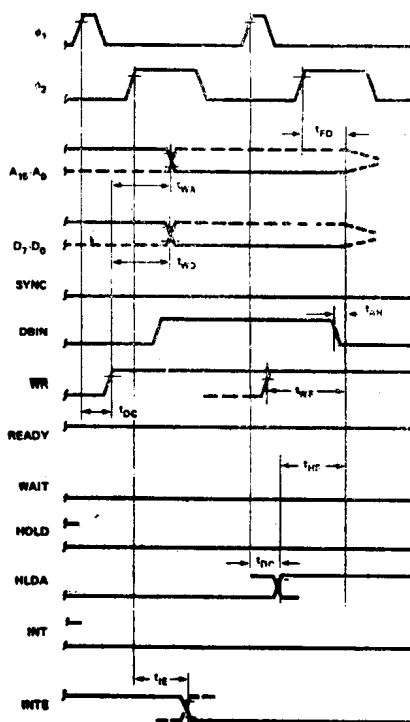
C_L = 100 pF
C_L INCLUDES JIG CAPACITANCE

波形



注意：时间的测量是按如下参改电压进行的；时钟“1”=8.0V，“0”=1.0V，输入“1”=3.3V，“0”=0.8V。输出“1”=2.0V，“0”=0.8V

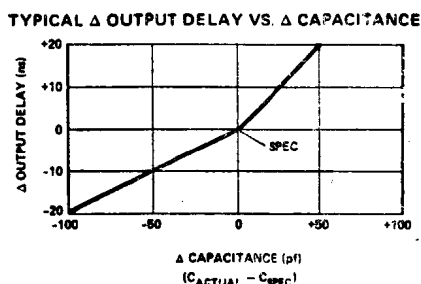
波形(继续)



注意：(括号分别指出 8080A-1, -2 按其详细说明)。

1. 用 DBIN 状态信号, 控制数据允许输入, 此后, 使总线不至于发生冲突, 保证了数据保持的时间。 $t_{DH} = 50ns$ 或 t_{DF} 无论哪一个都得要小。
2. $t_{CY} = t_{DB} + t_{r\phi 2} + t_{\phi 2} + t_{t\phi 2} + t_{D2} + t_{r\phi 1} \geq 480ns$ (-1 : 320ns, -2 : 380ns)

典型的 Δ 输出延迟与 Δ 容量的关系



3. 当 8080A 与具有 $V_{IH} = 3.3V$ 的设备接口时有下列关系:
 - a. 最大输出上升时间从 0.8V 至 3.3V = 100ns @ $C_L = SPEC$ (规定值)。
 - b. 当测量到 3.0 伏时输出延迟 = SPEC + 60ns @ $C_L = SPEC$ (规定值)。
 - c. $C_L = SPEC$ 规定值的话
 若 $C_L > C_{SPEC}$ 的话, 则加 0.6ns/pf
 若 $C_L < C_{SPEC}$ 的话, 则减去 0.3ns/pf (修改过的延迟)
4. $t_{AW} = 2t_{CY} - t_{D3} - t_{r\phi 2} - 140ns$ (-1 : 110ns, -2 : 130ns)
5. $t_{DW} = t_{CY} - t_{D3} - t_{r\phi 2} - 170$ (-1 : 150ns, -2 : 170ns)
6. 如果不是 HLDA, $t_{WD} = t_{WA} = t_{D3} + t_{r\phi 2} + 10ns$
 如果是 HLDA, $t_{WD} = t_{WA} + t_{WF}$
7. $t_{HF} = t_{D3} + t_{r\phi 2} - 50ns$
8. $t_{WF} = t_{D3} + t_{r\phi 2} - 10ns$
9. 在 DBIN T_3 期间, 对这段时间输入数据应该是稳定的, t_{DS1} 及 t_{DS2} 两者都应满足要求。
10. 在 T_2 或 T_W 期间, READY 信号在这段时间应该是稳定的, (必须是外同步)。
11. 当进入 HOLD 方式时在 T_2 或 T_W 期间, 及当在 Hold 方式, 在 T_1 、 T_4 、 T_3 和 T_{WH} 之期间, 对这段时间 HOLD 信号应该是稳定的 (不需要外同步)。
12. 任何指令的最后时钟周期的这段时间, 中断信号应该是稳定的。以便能被识别的后面指令 (不需要外同步)。
13. 这个时序图仅表示定时的关系, 并不表示任何具体的机器周期。

指令系统

累加器组的指令含有直接, 间接, 和立即寻址方式的算术和逻辑操作。

传送, 装入存取指令组能在存储器, 6 个工作寄存器和累加器之间用直接、间接, 及

立即寻址方式传送 8 位或 16 位的数据。

用转移, 条件转移及算术转移指令, 能提供各部分的程序转移, 也能提供条件和无条件子程序调用及返回。RESTART (或单字节调用指令) 用于中断向量的操作。

双倍精度操作, 例如堆栈操作和双字长加指令, 不仅扩充了 8080A 的算术运算而且还扩充了中断处理能力。能进行存贮器, 6 个通用寄存器, 累加器的加 1 及减 1 操作和所提供寄存器对及堆栈指示器加 1 及减 1 操作一样, 此外还能提供通过或不通过进位位累加器循环的左移或右移操作的能力。

输入及输出能进行存贮器映象 I/O 的寻址或 8080A 指令系统所提供的直接 I/O 寻址。

8080A 指令系统还有下列特殊指令组, NOP(空操作)指令, 停止执行存取的 HALT (停机)指令, 及能进行十进制数运算的 DAA(累加器的十进制调整)指令。允许直接置位进位标志的 STC(进位置位)指令, 累加器的内容变反码的 CMA(累加器变反)指令, 两个 16 位寄存器对的内容直接交换的 XCHG(寄存器对交换)指令。

数据和指令格式

数据在 8080A 中是以 8 位二进制整数形式存入的。所有数据以同样格式传送到系统数据总线。

D₇ D₆ D₅ D₄ D₃ D₂ D₁ D₀

数 据 字

程序指令长度可以是 1, 2 或 3 个字节, 多字节指令必须顺序存放在存贮单元里, 指令格式与所执行的具体操作有关。

单字节指令

D₇ D₆ D₅ D₄ D₃ D₂ D₁ D₀

操作码

典型指令

寄存器与寄存器, 存贮器, 算术或逻辑, 循环, 返回、入栈、出栈, 允许或禁止中断指令。

双字节指令

D₇ D₆ D₅ D₄ D₃ D₂ D₁ D₀

操作码

立即数寻址或 I/O 指令

D₇ D₆ D₅ D₄ D₃ D₂ D₁ D₀

操作数

三字节指令

D₇ D₆ D₅ D₄ D₃ D₂ D₁ D₀

操作码

转移, 调用或直接存取指令

D₇ D₆ D₅ D₄ D₃ D₂ D₁ D₀

低位地址或操作数 1

D₇ D₆ D₅ D₄ D₃ D₂ D₁ D₀

高位地址或操作数 2

8080A 逻辑“1”定义为高电平, 逻辑“0”定义为低电平

表 2 8080A 指令系统总表

Mnemonic	Instruction Code [1]								Operations Description	Clock Cycles [2]
	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀		
MOVE, LOAD, AND STORE										
MOV _{r1, r2}	0	1	D	D	D	S	S	S	Move register to register	5
MOV _{M, r}	0	1	1	1	0	S	S	S	Move register to memory	7
MOV _{r, M}	0	1	D	D	D	1	1	0	Move memory to register	7
MVI _r	0	0	D	D	D	1	1	0	Move immediate register	7
MVI _M	0	0	1	1	0	1	1	0	Move immediate memory	10
LXI _B	0	0	0	0	0	0	0	1	Load immediate register Pair B & C	10
LXI _D	0	0	0	1	0	0	0	1	Load immediate register Pair D & E	10
LXI _H	0	0	1	0	0	0	0	1	Load immediate register Pair H & L	10
STAX _B	0	0	0	0	0	0	1	0	Store A indirect	7
STAX _D	0	0	0	1	0	0	1	0	Store A indirect	7
LDAX _B	0	0	0	0	1	0	1	0	Load A indirect	7
LDAX _D	0	0	0	1	1	0	1	0	Load A indirect	7
STA	0	0	1	1	0	0	1	0	Store A direct	13
LDA	0	0	1	1	1	0	1	0	Load A direct	13
SHLD	0	0	1	0	0	0	1	0	Store H & L direct	16
LHLD	0	0	1	0	1	0	1	0	Load H & L direct	16
XCHG	1	1	1	0	1	0	1	1	Exchange D & E, H & L Registers	4
STACK OPS										
PUSH _B	1	1	0	0	0	1	0	1	Push register Pair B & C on stack	11
PUSH _D	1	1	0	1	0	1	0	1	Push register Pair D & E on stack	11
PUSH _H	1	1	1	0	0	1	0	1	Push register Pair H & L on stack	11
PUSH _{PSW}	1	1	1	1	0	1	0	1	Push A and Flags on stack	11
POP _B	1	1	0	0	0	0	0	1	Pop register Pair B & C off stack	10
POP _D	1	1	0	1	0	0	0	1	Pop register Pair D & E off stack	10
POP _H	1	1	1	0	0	0	0	1	Pop register Pair H & L off stack	10
POP _{PSW}	1	1	1	1	0	0	0	1	Pop A and Flags off stack	10
XTHL	1	1	1	0	0	0	1	1	Exchange top of stack, H & L	18
SPHL	1	1	1	1	1	0	0	1	H & L to stack pointer	5
LXI _{SP}	0	0	1	1	0	0	0	1	Load immediate stack pointer	10
INX _{SP}	0	0	1	1	0	0	1	1	Increment stack pointer	5
DCX _{SP}	0	0	1	1	1	0	1	1	Decrement stack pointer	5
JUMP										
JMP	1	1	0	0	0	0	1	1	Jump unconditional	10
JC	1	1	0	1	1	0	1	0	Jump on carry	10
JNC	1	1	0	1	0	0	1	0	Jump on no carry	10
JZ	1	1	0	0	1	0	1	0	Jump on zero	10
JNZ	1	1	0	0	0	0	1	0	Jump on no zero	10
JP	1	1	1	1	0	0	1	0	Jump on positive	10
JM	1	1	1	1	1	0	1	0	Jump on minus	10
JPE	1	1	1	0	1	0	1	0	Jump on parity even	10
JPO	1	1	1	0	0	0	1	0	Jump on parity odd	10
PCHL	1	1	1	0	1	0	0	1	H & L to program counter	5

续 表

Mnemonic	Instruction Code [1] D ₇ D ₆ D ₅ D ₄ D ₃ D ₂ D ₁ D ₀	Operations Description	Clock Cycles [2]
CALL			
CALL	1 1 0 0 1 1 0 1	Call unconditional	17
CC	1 1 0 1 1 1 0 0	Call on carry	11/17
CNC	1 1 0 1 0 1 0 0	Call on no carry	11/17
CZ	1 1 0 0 1 1 0 0	Call on zero	11/17
CNZ	1 1 0 0 0 1 0 0	Call on no zero	11/17
CP	1 1 1 1 0 1 0 0	Call on positive	11/17
CM	1 1 1 1 1 1 0 0	Call on minus	11/17
CPE	1 1 1 0 1 1 0 0	Call on parity even	11/17
CPO	1 1 1 0 0 1 0 0	Call on parity odd	11/17
RETURN			
RET	1 1 0 0 1 0 0 1	Return	10
RC	1 1 0 1 1 0 0 0	Return on carry	5/11
RNC	1 1 0 1 0 0 0 0	Return on no carry	5/11
RZ	1 1 0 0 1 0 0 0	Return on zero	5/11
RNZ	1 1 0 0 0 0 0 0	Return on no zero	5/11
RP	1 1 1 1 0 0 0 0	Return on positive	5/11
RM	1 1 1 1 1 0 0 0	Return on minus	5/11
RPE	1 1 1 0 1 0 0 0	Return on parity even	5/11
RPO	1 1 1 0 0 0 0 0	Return on parity odd	5/11
RESTART			
RST	1 1 A A A 1 1 1	Restart	11
INCREMENT AND DECREMENT			
INR r	0 0 D D D 1 0 0	Increment register	5
DCR r	0 0 D D D 1 0 1	Decrement register	5
INR M	0 0 1 1 0 1 0 0	Increment memory	10
DCR M	0 0 1 1 0 1 0 1	Decrement memory	10
INX B	0 0 0 0 0 0 1 1	Increment B & C registers	5
INX D	0 0 0 1 0 0 1 1	Increment D & E registers	5
INX H	0 0 1 0 0 0 1 1	Increment H & L registers	5
DCX B	0 0 0 0 1 0 1 1	Decrement B & C	5
DCX D	0 0 0 1 1 0 1 1	Decrement D & E	5
DCX H	0 0 1 0 1 0 1 1	Decrement H & L	5
ADD			
ADD r	1 0 0 0 0 S S S	Add register to A	4
ADC r	1 0 0 0 1 S S S	Add register to A with carry	4
ADD M	1 0 0 0 0 1 1 0	Add memory to A	7
ADC M	1 0 0 0 1 1 1 0	Add memory to A with carry	7
ADI	1 1 0 0 0 1 1 0	Add immediate to A	7
ACI	1 1 0 0 1 1 1 0	Add immediate to A with carry	7
DAD B	0 0 0 0 1 0 0 1	Add B & C to H & L	10
DAD D	0 0 0 1 1 0 0 1	Add D & E to H & L	10

续 表

Mnemonic	Instruction Code [1] D ₇ D ₆ D ₅ D ₄ D ₃ D ₂ D ₁ D ₀	Operations Description	Clock Cycles [2]
DAD H	0 0 1 0 1 0 0 1	Add H & L to H & L	10
DAD SP	0 0 1 1 1 0 0 1	Add stack pointer to H & L	10
SUBTRACT			
SUB r	1 0 0 1 0 S S S	Subtract register from A	4
SBB r	1 0 0 1 1 S S S	Subtract register from A with borrow	4
SUB M	1 0 0 1 0 1 1 0	Subtract memory from A	7
SBB M	1 0 0 1 1 1 1 0	Subtract memory from A with borrow	7
SUI	1 1 0 1 0 1 1 0	Subtract immediate from A	7
SBI	1 1 0 1 1 1 1 0	Subtract immediate from A with borrow	7
LOGICAL			
ANA r	1 0 1 0 0 S S S	And register with A	4
XRA r	1 0 1 0 1 S S S	Exclusive Or register with A	4
ORA r	1 0 1 1 0 S S S	Or register with A	4
CMP r	1 0 1 1 1 S S S	Compare register with A	4
ANA M	1 0 1 0 0 1 1 0	And memory with A	7
XRA M	1 0 1 0 1 1 1 0	Exclusive Or memory with A	7
ORA M	1 0 1 1 0 1 1 0	Or memory with A	7
CMP M	1 0 1 1 1 1 1 0	Compare memory with A	7
ANI	1 1 1 0 0 1 1 0	And immediate with A	7
XRI	1 1 1 0 1 1 1 0	Exclusive Or immediate with A	7
ORI	1 1 1 1 0 1 1 0	Or immediate with A	7
CPI	1 1 1 1 1 1 1 0	Compare immediate with A	7
ROTATE			
RLC	0 0 0 0 0 1 1 1	Rotate A left	4
RRC	0 0 0 0 1 1 1 1	Rotate A right	4
RAL	0 0 0 1 0 1 1 1	Rotate A left through carry	4
RAR	0 0 0 1 1 1 1 1	Rotate A right through carry	4
SPECIALS			
CMA	0 0 1 0 1 1 1 1	Complement A	4
STC	0 0 1 1 0 1 1 1	Set carry	4
CMC	0 0 1 1 1 1 1 1	Complement carry	4
DAA	0 0 1 0 0 1 1 1	Decimal adjust A	4
INPUT/OUTPUT			
IN	1 1 0 1 1 0 1 1	Input	10
OUT	1 1 0 1 0 0 1 1	Output	10
CONTROL			
EI	1 1 1 1 1 0 1 1	Enable Interrupts	4
DI	1 1 1 1 0 0 1 1	Disable Interrupt	4
NOP	0 0 0 0 0 0 0 0	No-operation	4
HLT	0 1 1 1 0 1 1 0	Halt	7

注意:

1. DDD或SSS: B=000, C=001, D=010, E=011, H=100, L=101, 存储器=110, A=111

2. 有两种可能的周期数(6/12)表示指令周期数和条件标志有关。

8085AH/8085AH-2/8085AH-1

8 位 HMOS 微处理器

- 单电源 +5 伏, 电压裕度 10%。
- 3MHz、5MHz、6MHz 三种频率可供选择
- 与 3MHz、5MHz 的 8085A 比较功率损耗降低 20%
- 指令周期为 1.3 μ s(8085AH); 0.8 μ s(8085AH-2); 0.67 μ s(8085AH-1)
- 与 8085A 完全兼容。
- 与 8080A 软件完全兼容。
- 片内时钟发生器(也可采用外接晶体, LC 或 RC 网络)
- 片内系统控制器; 对于大系统控制来说可用前置周期状态信息。
- 四个向量中断(其中之一为不可屏蔽中断)和一个同 8080A 兼容的中断。
- 串行输入/串行输出转接口。
- 十进制, 二进制和双倍精度运算。
- 对存储器 64K 字节有直接寻址能力

英特尔 8085AH 是一种完备的 8 位中央处理器(CPU), 以 N 沟道耗尽型硅栅工艺(HMOS)制作的。它的指令系统与 8080A 微处理器软件完全兼容, 并采用较高的系统速度改进现有的 8080A 的性能。它的系统集成度高, 允许由三块集成电路即 8085AH (CPU), 8156H(RAM/IO), 和 8355/8755(ROM/PROM/IO)组成一个最小系统而保持整个系统可扩展性, 8085AH-2 及 8085AH-1 是快速型式的 8085AH。

8085AH 把 8080A 所具备的 8224(时钟发生器)和 8228(系统控制器)所有功能都包括进去, 因此提高了本系统器件的集成度。

8085AH 采用多重数据总线, 把 16 位地址分为 8 位地址总线和 8 位地址/数据总线。8155H/8156H/8355/8755A 片内, 存储器的地址锁存器允许直接同 8085AH 相连。

8085AH CPU 功能框图如图 1 所示, 引脚排列如图 2 所示:

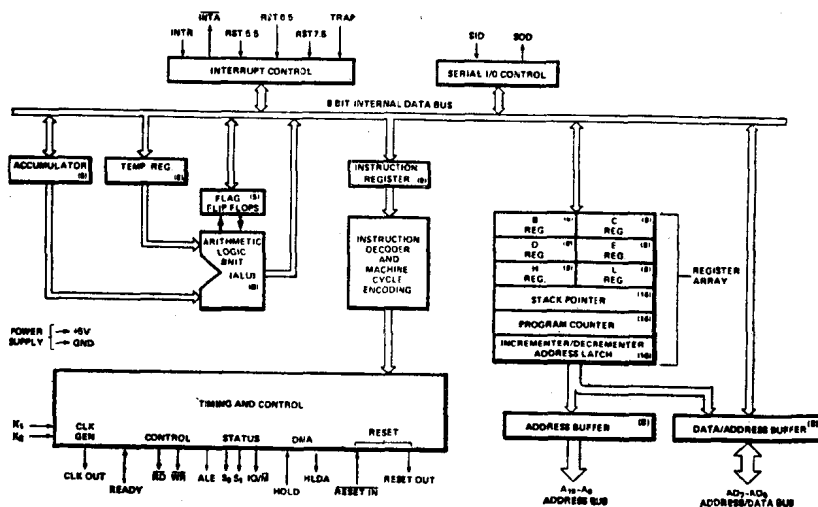


图 1 8085AH CPU 功能框图

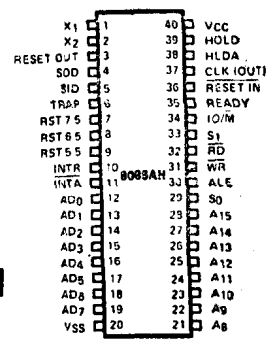


图 2 8085AH 引脚图

表 1 8085AH 引脚说明

符 号	类 型	名 称 及 功 能																																								
$A_8 - A_{15}$	O	Address Bus(地址总线): 有效的高8位存储器地址或8位输入/输出地址, 在HOLD(保持)或HALT(停机)方式及 RESET(复位)时为3态输出。																																								
$AD_0 - 7$	I/O	Multiplexed Address/Data Bus(多重地址/数据总线): 在机器周期的第一个时钟周期(状态 T_1)时存储器低8位地址(或I/O地址)出现在总线上, 然后在第二和第三时钟周期(T_2 和 T_3)时作为数据总线。																																								
ALE	O	Address Latch Enable(地址允许锁存): 该信号出现在机器周期的第一时钟周期(状态 T_1)并允许地址锁存到外设片内锁存器。用ALE信号下降沿去保证建立并保持地址信息, ALE下降沿也能用来选通状态信息, ALE绝不作为3态信号。																																								
S_0, S_1 和 $IO/\overline{M}$	O	Machine cycle status(机器周期状态): <table><tr><th>$IO/\overline{M}$</th><th>S_1</th><th>S_0</th><th>Status</th></tr><tr><td>0</td><td>0</td><td>1</td><td>存储器写</td></tr><tr><td>0</td><td>1</td><td>0</td><td>存储器读</td></tr><tr><td>1</td><td>0</td><td>1</td><td>I/O 写</td></tr><tr><td>1</td><td>1</td><td>0</td><td>I/O 读</td></tr><tr><td>0</td><td>1</td><td>1</td><td>取操作码</td></tr><tr><td>1</td><td>1</td><td>1</td><td>中断响应</td></tr><tr><td>•</td><td>0</td><td>0</td><td>停机</td></tr><tr><td>•</td><td>×</td><td>×</td><td>保持</td></tr><tr><td>•</td><td>×</td><td>×</td><td>复位</td></tr></table> • = 3态(高阻抗状态) × = 不确定状态 S_1 可用作前置的 $R/\overline{W}$ 状态信号。$IO/\overline{M}$, S_0 和 S_1 在一个机器周期开始时就有效, 并在整个周期稳定不变, ALE 信号的下降沿可用来锁存这些信号的状态。	$IO/\overline{M}$	S_1	S_0	Status	0	0	1	存储器写	0	1	0	存储器读	1	0	1	I/O 写	1	1	0	I/O 读	0	1	1	取操作码	1	1	1	中断响应	•	0	0	停机	•	×	×	保持	•	×	×	复位
$IO/\overline{M}$	S_1	S_0	Status																																							
0	0	1	存储器写																																							
0	1	0	存储器读																																							
1	0	1	I/O 写																																							
1	1	0	I/O 读																																							
0	1	1	取操作码																																							
1	1	1	中断响应																																							
•	0	0	停机																																							
•	×	×	保持																																							
•	×	×	复位																																							
$\overline{RD}$	O	Read Control(读控制): $\overline{RD}$ 为低电平, 表示从所选择的存储器或输入/输出装置读出来, 并表示数据总线可用于数据传送。 在 Hold (保持)和 Halt (停机)方式及 RESET(复位)时为 3 态输出。																																								
$\overline{WR}$	O	Write Control(写控制): $\overline{WR}$ 为低电平, 表示数据总线上的数据写入到所选择的存储器或输入/输出地址里, 在 $\overline{WR}$ 信号的后沿建立数据, 在 Hold 和 Halt 方式及 RESET 时为 3 态输出。																																								
READY	I	Ready(准备就绪): 如果在读或写周期时 READY(准备就绪)信号为高电平, 就表示存储器或 I/O 外设准备好发送或接收数据。如果 READY 信号为低电平, 那么CPU在完成读或写周期之前将等待一个整数时钟周期直到 READY信号变为高电平为止。READY 应该与建立和保持时间一致。																																								
HOLD	I	Hold(保持请求): 表示其它主设备需要使用地址或数据总线, 若CPU接受保持请求, 一旦现行总线传送完成时放弃使用总线。中央处理器仍能继续工作, 只有在 HOLD 信号撤除之后, 处理器方能再度获得总线。当 HOLD 信号被响应时, 地址, 数据, $\overline{RD}$, $\overline{WR}$ 及 $IO/\overline{M}$ 等信号皆处于 3 态。																																								

续 表

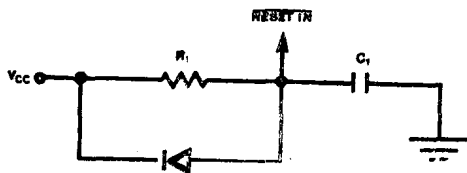
HLDA	O	Hold Acknowledge(保持响应): 表示CPU已经接受 HOLD请求, 并将在下一个时钟周期放弃总线。在HOLD保持请求信号撤除之后, HLDA变成低电平状态。在 HLDA 变成低电平之后半时钟周期 CPU 获得总线。
INTR	I	Interrupt Request (中断请求): 用作一般中断, 仅在指令的最后一个时钟周期被采样, 若该信号有效, 则程序计数器 (CP) 将停止计数, 并发出 $\overline{\text{INTA}}$ 信号, 在这期间插入 RESTART (重入启动) 或 CALL (调用) 指令以便转移到中断服务程序。由软件来决定是否允许中断或禁止中断。复位和在一个中断被接收后立即被禁止。
$\overline{\text{INTA}}$	O	Interrupt Acknowledge(中断响应): 中断请求 (INTR) 被接收之后, 在指令周期内用 $\overline{\text{INTA}}$ 来替代 $\overline{\text{RD}}$ (具有相同时间), 并能启用 8259A 中断接口器件和其他一些中断转接口。
RST5.5 RST6.5 RST7.5	I	Restart Interrupts (重入中断): 除自动转入一个内部重新启动信号外这三个中断输入同 INTR 中断信号具有同样时序。这些中断的优先级次序如表 2 所示, 具有比 INTR 更高的优先级。另外, 可用 SIM 指令对它们分别屏蔽。
TRAP	I	Trap (自陷中断): 自陷中断是一个不可屏蔽的重入启动中断, 它与 INTR 或 RST5.5-7.5 有同样时序。它不受任何屏蔽或允许中断信号影响, 具有所有中断输入中的最高优先级。(见表 2)
$\overline{\text{RESET IN}}$	I	Reset IN(复位输入): 置程序计数器(PC)为零, 并清除中断允许和停机(HLDA)触发器。在复位时, 数据和地址总线及控制信号处于三态, 由于 RESET (复位) 具有异步特性, 可使处理器的内部寄存器和标志寄存器为任意值。 $\overline{\text{RESET IN}}$ 为斯密特触发器输入, 允许同加电复位延迟的 RC 网络相连(见图 3)。在加上电源达到最小 V_{cc} 值之后 $\overline{\text{RESET IN}}$ 必须保持 10ms 低电平, 在加电过程中为了得到适当的复位操作, $\overline{\text{RESET IN}}$ 应保持最小是三个时钟周期的低电平。只要施加 $\overline{\text{RESET IN}}$, CPU 就保持在复位状态。
RESET OUT	O	Reset Out(复位输出): 复位输出表示 CPU 正在清除, 能用作系统总清。该信号与 CPU 时钟脉冲同步, 并持续时钟周期的一个整数倍。
X_1, X_2	I	X_1 and X_2 : 与晶体, LC 或 RC 网络连结的驱动片内时钟发生器, X_1 也能是一个来自逻辑门电路输入外部时钟脉冲输入端。其输入频率为处理器片内工作频率的 2 倍。
CLK	O	Clock(时钟): 时钟输出用作系统时钟, 时钟周期为 X_1, X_2 输入周期的 2 倍。
SID	I	Serial Input Data Line(串行输入数据线): 每当执行一条 RIM 指令时, 此线上的数据就打入到累加器第七位。
SOD	O	Serial Output Data Line(串行输出数据线): 按照 SIM 指令来指明 SOD 是置“1”或置“0”。
V_{cc}		Power(电源): +5 伏。
V_{ss}		Ground(地): 基准地线

表 2 中断优先级重入地址及触发方式

名 称	优先级	产生中断时转移地址(注1)	触 发 类 型
TRAP	1	24H	前沿触发并维持高电平直到接受
RST7.5	2	3CH	前沿触发(锁存)
RST6.5	3	34H	高电平直到接收
RST5.5	4	2CH	高电平直到接受
INTR	5	见注释(2)	高电平直到接受

注释: 1. 在转移到给定地址前处理器把 PC 压入堆栈

2. 当中断被响应时, 转移地址由CPU提供的指令所决定。



TYPICAL POWER-ON RESET RC VALUES*
 $R_1 = 75 \text{ K}\Omega$
 $C_1 = 1 \mu\text{F}$
 *VALUES MAY HAVE TO VARY DUE TO
 APPLIED POWER SUPPLY RAMP UP TIME

图 3 加电复位电路

功能说明:

8085AH是一种完备的8位中央处理器, 它采用N—沟道耗尽型HMOS工艺制造的, 只需+5伏单电源。8085AH基准时钟脉冲频率为3MHz。8085AH-2为5MHz, 8085AH-1为6MHz。因此它是在现有的8080A的性能上加以改进采用较高的系统速度, 只用三块集成电路即一片CPU(8085AH), 一片RAM/IO(8156H)和一片ROM或EPROM(8355或8755A), 可组成一个最小系统。

8085AH有12个可寻址的8位寄存器, 其中4个仅可成对组成2个16位寄存器对, 另外6个能用作可互换的8位寄存器或16位寄存器对。8085AH寄存器组如下所示:

名 称	寄 存 器	内 容
ACC 或 A	累加器	8 位
PC	程序计数器	16位地址
BC, DE, HL	通用寄存器,	8位×6
	数据指示器(HL),	或16位×3
SP	堆栈指示器	16位地址
Flags 或 F	标志寄存器	5个标志(8位空间)。

8085AH使用多重数据总线, (multiplexed Data Bus), 把地址分为高8位地址总线和低8位地址/数据总线在机器周期的第一时钟周期(T_1 状态)把低位地址发送到地址/数据总线, 由地址允许锁存信号(ALE)把这些低8位地址在外部锁存器锁存起来, 在以后的机器周期(T_2, T_3)数据总线为存储器或输入/输出设备数据所使用。

8085AH为总线控制提供 $\overline{\text{RD}}$, $\overline{\text{WR}}$, S_0 , S_1 及 $\text{IO}/\overline{\text{M}}$ 信号。还提供一个中断响应信号(INTA)。HOLD(保持请求)及所有中断与处理器内部时钟是同步的。8085AH还为简单串行接口提供串行数据输入(SID)和串行数据输出(SOD)信号。

除这些特点之外, 8085AH有三个可屏蔽的向量中断, 一个不可屏蔽中断 TRAP 和一

个总线向量中断请求信号 INTR

中断和串行输入/输出

8085AH有5个中断输入：INTR RST5.5 RST6.5 RST7.5 和TRAP。INTR与8080AH INT的功能相同，三个重入中断(RESTART)RST5.5、6.5和7.5的每一个都是可用程序屏蔽的，TRAP 也是一个重入中断，但它是不可屏蔽的。

如果中断允许，中断屏蔽位不设置屏蔽则这三种可屏蔽中断产生内部重入中断控制信号(程序计数器的内容送入堆栈，并转移到重入中断地址)。不可屏蔽 TRAP 中断为产生与中断允许或中断屏蔽无关的另一重入中断控制信号。(见表 2)。

就重入中断而言，有二个不同类型的输入，RST5.5 和 RST6.5 像 INTR (和 8080 的 INT)一样是高电平起作用，也用与 INTR 相同的时钟信号。RST7.5 为上升沿起作用，对于 RST7.5 来说，只需要一个请求脉冲对产生内部中断请求的内部触发器置位(通常是高电平信号，对于系统抗干扰要求很高的系统可推荐用低电平触发)。RST7.5 请求中断触发器保持置“1”状态直到请求获准服务为止。然后自动复位触发器，这个触发器也能用 SIM 指令或用8085AH复位输入信号(RESET IN)来复位。当 RST7.5 中断屏蔽输出时，RST7.5 的触发器也可用 RST7.5 引脚的一个脉冲信号来置位。影响三个 RST (重入中断) 中断屏蔽码的状态只能用 SIM 指令和复位输入信号(RESET IN)(见 MCS-80/85使用手册第 5 章 SIM 指令)。

中断信号按一定的优先级排列。如果多个中断请求，哪一个中断被识别，优先级规定的次序如下：TRAP—最高的优先级，其次是 RST7.5，RST6.5，RST5.5，INTR—最低的优先级。这个优先级分类表不考虑较高优先级中断开始的程序的中断优先级，如果 RST7.5 中断程序结束之前中断被开中，那么 RST5.5能中断 RST7.5 的程序。

TRAP (陷阱中断)对于例如电源失效或总线出错等严重故障情况是有用的，TRAP 中断输入可像任何其它中断一样被识别，但它具有最高的优先级它并不影响任何标志位和屏蔽码，TRAP 中断输入信号的脉冲沿和电平信号二者都有才起作用。TRAP 中断输入必须达到高电平并保持高电平直到被响应，它变为低电平不再被识别，然后变为高电平再被识别，这就避免由于干扰或逻辑线路尖峰而引起任何假触发信号。图 4 说明在 8085AH 里所用的 TRAP 中断请求电路。应该注意：正在服务的任何中断 (TRAP, RST7.5, RST6.5, RST5.5, INTR)都将禁止其它中断(除 TRAP 外)直到 EI 指令被执行为止。

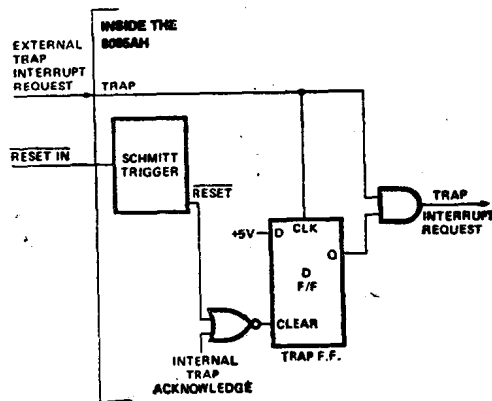


图 4 TRAP 和 RESET IN 电路