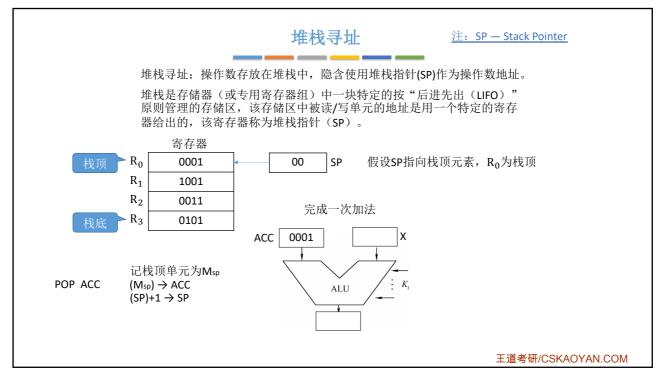


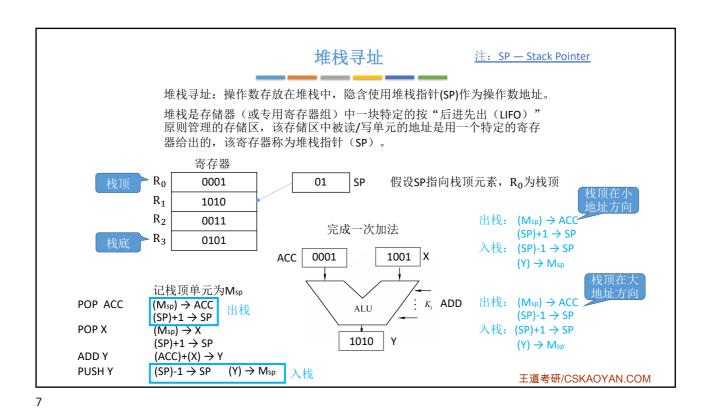
## 堆栈寻址 注: SP — Stack Pointer 堆栈寻址:操作数存放在堆栈中,隐含使用堆栈指针(SP)作为操作数地址。 堆栈是存储器(或专用寄存器组)中一块特定的按"后进先出(LIFO)" 原则管理的存储区,该存储区中被读/写单元的地址是用一个特定的寄存 器给出的,该寄存器称为堆栈指针(SP)。 寄存器 $R_0$ 0001 SP 假设SP指向栈顶元素, $R_0$ 为栈顶 $R_1$ 1001 $R_2$ 0011 $R_3$ 0101

王道考研/CSKAOYAN.COM



## 堆栈寻址 注: SP — Stack Pointer 堆栈寻址:操作数存放在堆栈中,隐含使用堆栈指针(SP)作为操作数地址。 堆栈是存储器(或专用寄存器组)中一块特定的按"后进先出(LIFO)" 原则管理的存储区,该存储区中被读/写单元的地址是用一个特定的寄存 器给出的,该寄存器称为堆栈指针(SP)。 寄存器 $R_0$ 0001 01 SP 假设SP指向栈顶元素, $R_0$ 为栈顶 $R_1$ 1001 $R_2$ 0011 完成一次加法 $R_3$ 0101 0001 1001 X ACC 记栈顶单元为Msp $(M_{sp}) \rightarrow ACC$ $(SP)+1 \rightarrow SP$ POP ACC ALU POP X $(M_{sp}) \rightarrow X$ $(SP)+1 \rightarrow SP$ 王道考研/CSKAOYAN.COM

堆栈寻址 注: SP — Stack Pointer 堆栈寻址:操作数存放在堆栈中,隐含使用堆栈指针(SP)作为操作数地址。 堆栈是存储器(或专用寄存器组)中一块特定的按"后进先出(LIFO)" 原则管理的存储区,该存储区中被读/写单元的地址是用一个特定的寄存 器给出的,该寄存器称为堆栈指针(SP)。 寄存器 - R<sub>0</sub> 0001 SP 假设SP指向栈顶元素, $R_0$ 为栈顶 10  $R_1$ 1001  $R_2$ 0011 完成一次加法  $R_3$ 0101 0001 ACC 1001 X 记栈顶单元为Msp K. ADD POP ACC  $(M_{sp}) \rightarrow ACC$ ALU  $(SP)+1 \rightarrow SP$ POP X  $(M_{sp}) \rightarrow X$ 1010 Υ  $(SP)+1 \rightarrow SP$ ADD Y  $(ACC)+(X) \rightarrow Y$ **PUSH Y**  $(SP)-1 \rightarrow SP$ 王道考研/CSKAOYAN.COM



堆栈寻址 注: SP — Stack Pointer 堆栈寻址:操作数存放在堆栈中,隐含使用堆栈指针(SP)作为操作数地址。 堆栈是存储器(或专用寄存器组)中一块特定的按"后进先出(LIFO)" 原则管理的存储区,该存储区中被读/写单元的地址是用一个特定的寄存 器给出的,该寄存器称为堆栈指针(SP)。 主存 寄存器  $R_0$ 0001 01 SP  $R_1$ 1010  $R_2$ 0011  $R_3$ 0101 0100110001 硬堆栈 软堆栈 堆栈可用于函数调用时保存当前函数的相关信息(可参考数据结构"算法空间复杂度"的视频) 王道考研/CSKAOYAN.COM

	_	本节回顾	
	寻址方式	有效地址	访 存 次 数(指令执行期间)
偏移寻址	隐含寻址	程序指定	0
	立即寻址	A即是操作数	0
	直接寻址	EA=A	1
	一次间接寻址	EA=(A)	2
	寄存器寻址	EA=R <sub>i</sub>	0
	寄存器间接一次寻址	EA=(R <sub>i</sub> )	1
	转移指令 相对寻址	EA=(PC)+A	1
	多道程序 基址寻址	EA=(BR)+A	1
	循环程序 变址寻址 数组问题	EA=(IX)+A	1
	堆栈寻址	入栈/出栈时EA的确定方式不同	硬堆栈不访存,软堆栈访存1次
			王道考研/CSKAOYAN