

第三次作业

程远 2234412848

1 第一题

我将整个加法过程分为以下五步（纸带起始位置为左起第一个非空数据）：

表 1: 启动程序，找到加号

当前状态为	纸带数据为	改写纸带为	改写状态为	指针移动为
STA	0	0	STA	R
STA	1	1	STA	R
STA	+	+	$F_{J_0 P_0}$	L

表 2: 找到对应位并进行加法

当前状态为	纸带数据为	改写纸带为	改写状态为	指针移动为
$F_{J_i P_j}$	X	X_C	$X_{J_i P_j}$	R
$F_{J_i P_j}$	X_C	X_C	$F_{J_i P_j}$	R
$X_{J_i P_j}$	X'	X'	$X_{J_i P_j}$	R
$X_{J_i P_0}$	+	+	$X_{J_i P_1}$	R
$F_{J_i P_0}$	+	+	$F_{J_i P_1}$	R
$X_{J_i P_1}$	=	=	$X_{J_i P_2}$	L
$F_{J_i P_1}$	=	=	$F_{J_i P_2}$	L
$X_{J_i P_2}$	X'	X'_C	$X''_{J_i' T}$	L
$F_{J_i P_2}$	X	X_C	$(X \text{ XOR } i)_{J_{(X \text{ XOR } i)} T}$	L
$X_{J_i P_2}$	+	+	$X_{J_i' T}$	L

$$X'' = (X \text{ XOR } X') \text{ XOR } i$$

$$i' = (X \text{ AND } X') \text{ OR } (X \text{ AND } i) \text{ OR } (X' \text{ AND } i)$$

表 3: 将结果传递至输入纸带左边

当前状态为	纸带数据为	改写纸带为	改写状态为	指针移动为
$X_{J_i T}$	X'	X'	$X_{J_i T}$	L
$X_{J_i T}$	X'_C	X'_C	$X_{J_i T}$	L
$X_{J_i T}$	+	+	$X_{J_i T}$	L
$X_{J_i T}$	NULL	X_T	$F_{J_i P_0}$	R
$X_{J_i T}$	X'_T	X'_T	$X_{J_i T}$	L

表 4: 处理最后一次位运算

当前状态为	纸带数据为	改写纸带为	改写状态为	指针移动为
$F_{J_i P_2}$	=	=	$F_{J_i E}$	L
$F_{J_i E}$	X_C	X	$F_{J_i E}$	L
$F_{J_i E}$	+	+	$F_{J_i E}$	L
$F_{J_i E}$	=	=	$F_{J_i E}$	L
$F_{J_1 E}$	NULL	1_T	M_f	N
$F_{J_0 E}$	NULL	NULL	M_f	N

表 5: 将输入左边的运算结果移动到等号右边, 并处理格式

当前状态为	纸带数据为	改写纸带为	改写状态为	指针移动为
M_f	X_T	NULL	M_f	R
M_f	X	X	M_{fx}	R
M_{fx}	+	+	M_{fx}	R
M_{fx}	=	=	M_{fx}	R
M_{fx}	NULL	X	M_b	L
M_b	X	X	M_b	L
M_b	+	+	M_b	L
M_b	=	=	M_b	L
M_b	X_T	X_T	M_b	L
M_b	NULL	NULL	M_C	R
M_C	X_T	NULL	M_{fx}	R
M_C	X	X	E	R
E	X	X	M_{fx}	R
E	+	+	M_{fx}	R
E	=	=	M_{fx}	R
E	NULL	b	ABORT	N

以下是一些注释：

STA 表示起始状态。

F 表示寻找未计算数状态。

X 表示状态时，为找到未计算数状态，表示数字时，为 0 或 1。

J_i 为进位器 (J_0 表示无进位, J_1 表示有进位), 下标 C 表示将该数标记为已计算,

P_i 为位置指示器 (P_0 、 P_1 、 P_2 分别表示第一次经过 “+”、第一次经过 “=”、第二次经过 “+”)

T 为搬运指示器, 指示图灵机将状态对应的数字添加到纸带左端。

M 状态为移动状态, 指示图灵机将在左端储存的计算结果移动到等号右端。

下标 f 表示向右移动, 下标 b 表示向左移动, 下标 x 表示当前移动的数。

E 状态为收尾状态, 指示图灵机进行调整格式的任务。

NULL 表示当前纸带位置上没有数据。

ABORT 表示停机。

2 第二题

现代计算机与冯·诺依曼计算机在结构上十分相似。冯·诺依曼结构由五部分组成, 包括输入设备、运算器、存储器、控制器和输出设备。基本符合现代计算机的组成, 即输入设备对应键鼠等设备; 运算器与控制器对应 CPU、GPU 甚至 NPU 等设备; 存储器对应内存、输出设备对应显示器、扬声器等设备。由此观之、现代计算机本质上没有跳出冯·诺依曼结构、现代计算机也都可以称为冯·诺依曼计算机。

计算机区分程序与数据有两种方法, 通过时间段区分与通过地址来源区分。一个指令的执行通常可以分为多个阶段, 包括取指阶段、分析阶段和执行阶段。由于 CPU 在取指阶段读取指令 (程序), 在执行阶段读取操作数 (数据), 所以可以通过当前位于取指阶段还是执行阶段来判断读取的内容是程序还是数据。除了通过时间段区分, 计算机还可以通过地址来源来区分程序与数据: 通过程序计数器提供的地址来访问内存获取的数据是指令 (程序), 在执行阶段由指令的地址码部分提供存储单元地址的数据是操作数 (数据), 不同的地址来源便对应不同的信息类型。