

CH7

Control Instructions:

BR TRAP JMP JSR RTI

condition code比起比较后再跳转的方式比较好，因为不需要额外的cost就实现了Branch所需要的东西。

Assembly Language

```
Label Opcode Operands ; comment
```

- 让程序员简化记忆机器码的复杂，但是仍然接近底层并且能够更好的调用系统资源。
- opcode + operands (每个指令的operands数量不同)
 - They are **mandatory** (必须的)
 - #代表十进制数（不写默认是10进制），x代表16进制，b代表2进制
 - ADD有三个operands，LD有两个operands，BR有一个operands，
- **Comment:** ; *semicolon*
- **Labels**
 - 提供了简便的计算地址的方式
 - 如果太远会怎么样？汇编器会提示失败
- **Pseudo-Ops** (与 #pragma 类似，是一些预编译指令):伪操作，并不是真正的指令
 - `.ORIG`声明了下一条指令的位置，它自己并不会占去一个位置（一个程序可以有多个`.ORIG`和`.END`配套）。默认的位置为`x3000`
 - `.END`声明了一段代码的结束（必须和`.ORIG`配套），之后的代码将被直接忽略
 - `.FILL` 由程序声明了一个空间用于存储一个16位的二进制数，The value can be either a **number** or a **label**.
 - `.BLKW` a **BLock** of **Words** 比起`.FILL`通常用于这段空间具体值未知的情况。
`.BLKW ?`表示预留了? 个地址空间，它们的首地址为`.BLKW`所在的地址
 - `.STRINGZ`声明了一段字符串，但请注意末尾的0，它所占的空间实际上是**字符串长度+1**
`HELLO .STRINGZ "Hello, world!"` 实际所占的位置为:

```
x3010: x0048
x3011: x0065
x3012: x006C
x3013: x006C
x3014: x006F
x3015: x002C
x3016: x0020
x3017: x0057
x3018: x006F
x3019: x0072
x301A: x006C
x301B: x0064
x301C: x0021
x301D: x0000
```

- *. *EXTERNAL*声明了外部的变量，可以显式的引用其他程序的label

The Assembly Process

所有程序在被执行之前都要被翻译成对应的机器语言，You remember that there is in general a **one-to-one** correspondence between instructions in an assembly language program and instructions in the final machine language program.

- **The First Pass: Creating the Symbol Table**

计算时请忽略单独的comment行，因为每个地址应该存放一个16位binary number

- **The Second Pass: Generating the Machine Language Program**

State Machine

702

appendix C The Microarchitecture of the LC-3

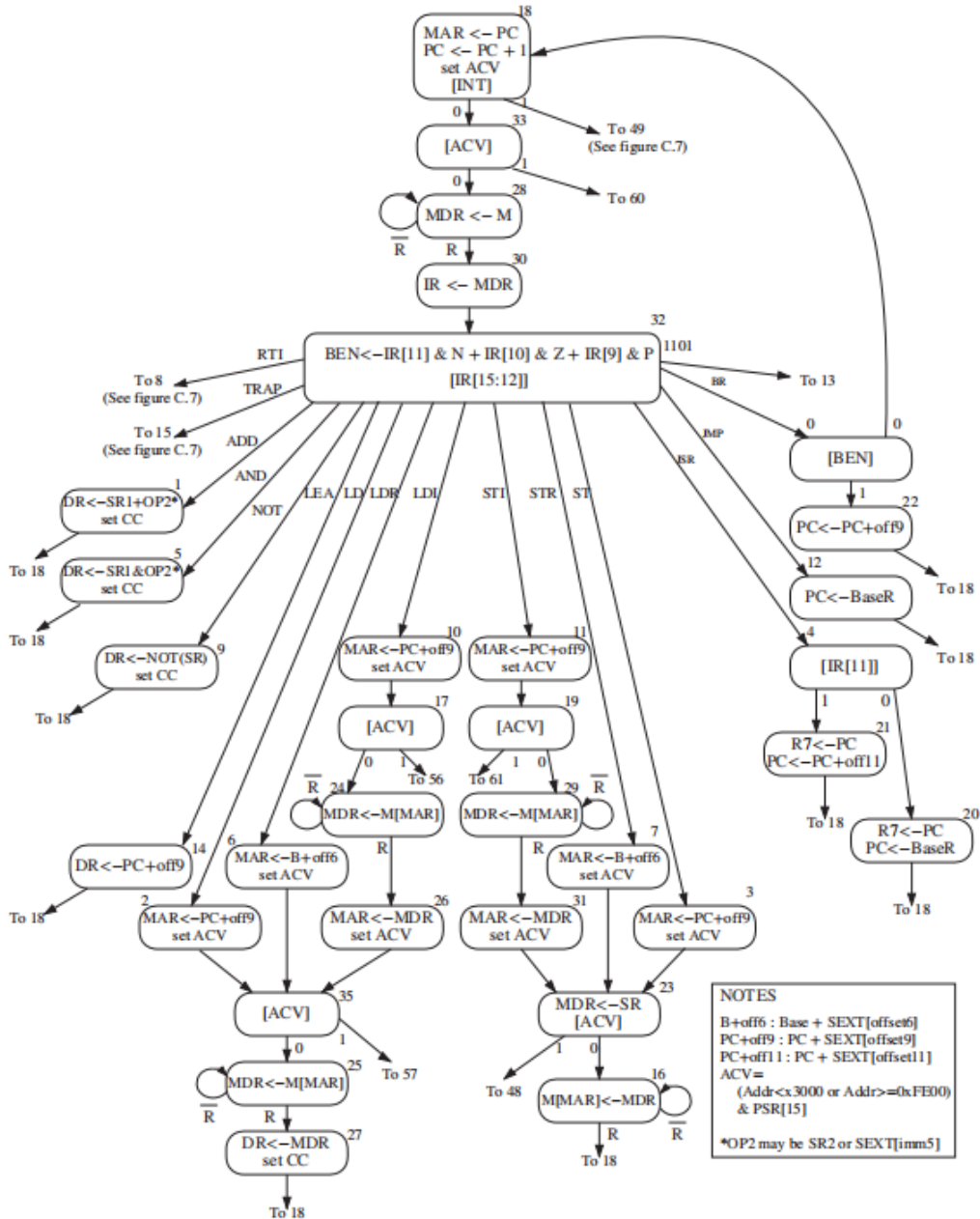


Figure C.2 A state machine for the LC-3.