

# 计算机组成原理 Homework14 (12.9)

中国人民大学 信息学院 崔冠宇 2018202147

1. 完成 CPU 指令的分解。根据课堂 ppt 中的 CPU 完整数据通路图，完成以下述指令的分解过程，指出含几个机器周期，以及每个机器周期中的控制信号。

(1) `push ri`

压栈指令，将寄存器 `ri` 的内容压入堆栈。

(2) `mov ri, [[rj + disp]]`

存储器间接寻址，将地址 `rj+disp` 指出的存储单元内容作为地址再次访问存储器，读取的结果送寄存器 `ri`。

(3) `call cc @disp`

条件过程调用，根据 `cc` 条件，如果成立，则实现通过对 `PC` 的相对转移偏移 `disp` 处的子程序调用，否则，进入下一条指令。

解：

(1) `push ri`，共需要 4 个机器周期：

1. 取指：

```
1 PC -> AB;
2 M/IO# = 1;
3 W/R# = 0;
4 DB -> IR;
5 PC+1;
```

2. 准备栈指针：

```
1 sp -> ALU;
2 '+' -> ALU;
3 ALU -> AR;
4 sp-1;
```

3. 准备数据：

```
1 ri -> GR;
2 (ri) -> ALU;
3 '+' -> ALU;
4 ALU -> DR;
```

4. 压栈：

```
1 AR -> AB;
2 DR -> DB;
```

```
3 M/IO# = 1;  
4 W/R# = 1;
```

(2) mov ri, [[rj + disp]], 共需要 6 个机器周期:

1. 取指: (与上面指令的取指周期完全相同, 略)

2. 准备地址:

```
1 rj -> GR;  
2 (rj) -> ALU;  
3 disp -> ALU;  
4 '+' -> ALU;  
5 ALU -> AR;
```

3. 第一次访存:

```
1 AR -> AB;  
2 M/IO# = 1;  
3 W/R# = 0;  
4 DB -> DR;
```

4. 准备地址:

```
1 DR -> ALU;  
2 '+' -> ALU;  
3 ALU -> AR;
```

5. 第二次访存:

```
1 AR -> AB;  
2 M/IO# = 1;  
3 W/R# = 0;  
4 DB -> DR;
```

6. 回送结果:

```
1 DR -> ALU;  
2 '+' -> ALU;  
3 ri -> GR;  
4 ALU -> GR;
```

(3) `call cc @disp`, 根据条件 `cc` 成立与否, 共需要 1 或 5 个机器周期:

1. 取指周期: (与上面指令的取指周期完全相同, 略)

若条件 `cc` 不成立, 则指令结束。

2. 准备栈指针:

```
1 sp -> ALU;  
2 '+' -> ALU;  
3 ALU -> AR;  
4 sp-1;
```

3. 准备栈数据:

```
1 PC -> ALU;  
2 '+' -> ALU;  
3 ALU -> DR;
```

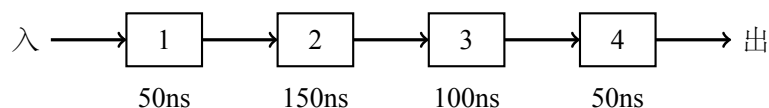
4. 压栈:

```
1 AR -> AB;  
2 DR -> DB;  
3 M/IO# = 1;  
4 W/R# = 1;
```

5. 计算转移地址:

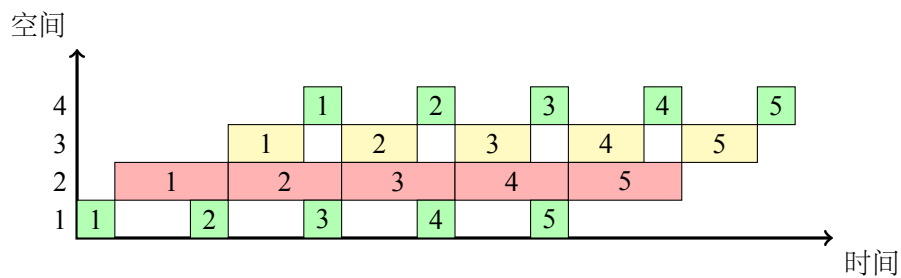
```
1 PC -> ALU;  
2 disp -> ALU;  
3 '+' -> ALU;  
4 ALU -> PC;
```

2. 某流水线共分为 4 段, 各段的所花的时间如下图所示:



请画出连续执行 5 条指令的时空图, 求解该流水线的实际吞吐率、最大吞吐率、加速比和效率。你觉得该流水线的瓶颈在哪一段?

解: 时空图如下:



在本例中， $n = 5$ ， $m = 4$ ， $\Delta t = 50\text{ns} = 5 \times 10^{-8}\text{s}$ ，总时间  $T_m = 3(n - 1)\Delta t + (1 + 3 + 2 + 1)\Delta t = 19\Delta t$ 。

(1) 吞吐率  $T_p = \frac{n}{T_m} = \frac{5}{19\Delta t} = \frac{5}{19 \times 5 \times 10^{-8}} = 5.26 \times 10^6$  指令/秒；

(2) 最大吞吐率  $T_{pmax} = \lim_{n \rightarrow +\infty} \frac{n}{T_m} = \lim_{n \rightarrow +\infty} \frac{n}{(3n+4)\Delta t} = \frac{1}{3\Delta t} = \frac{1}{3 \times 5 \times 10^{-8}} = 6.67 \times 10^6$  指令/秒；

(3) 串行方式时间  $T_0 = (1 + 3 + 2 + 1)n\Delta t = 35\Delta t$ ，因此加速比  $S = \frac{T_0}{T_m} = \frac{35\Delta t}{19\Delta t} = \frac{35}{19} = 1.84$ ；

(4) 效率  $\eta = \frac{\text{有任务格数}}{\text{总格数}} = \frac{(1+3+2+1) \times 5}{19 \times 4} = \frac{35}{76} = 46.05\%$ 。

显然，瓶颈在第二步。