NJUSWI
2002-2012
Anniversary
南京大學 软件学院

分解与抽象

刘 钦

南京大学软件学院

*

//打印1颗*

```
System.out.print(‘‘*’’);
```

#

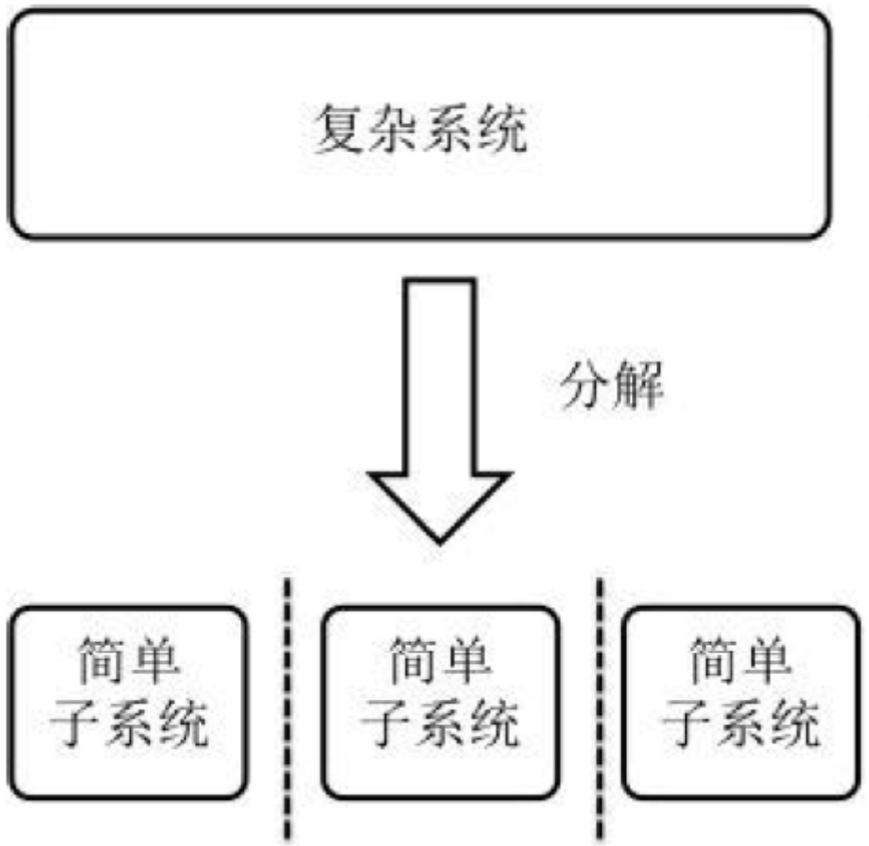
//打印1颗#

```
System.out.print(“#”);
```

```
*#
//  打印＊＃
//=打印一颗* + 打印一颗#
```

```
System.out.print(“*”);
System.out.print(“ # ”);
```

降低复杂度的方法 —分解



分解

分解的关键点

- 分解之后，每一部分复杂度要变小，
- 相互之间关联要小，相对独立。

FIGURE 7.1 The design of a computer.

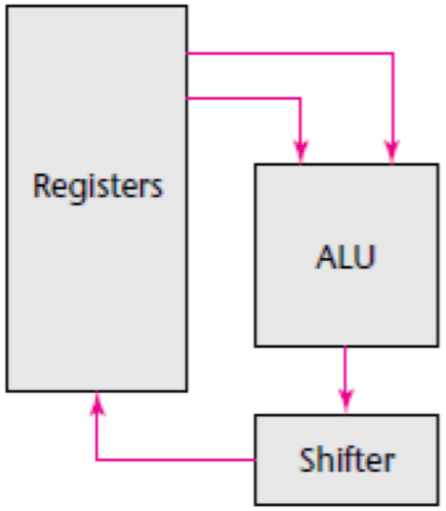


FIGURE 7.2 The computer of Figure 7.1 fabricated on three chips.

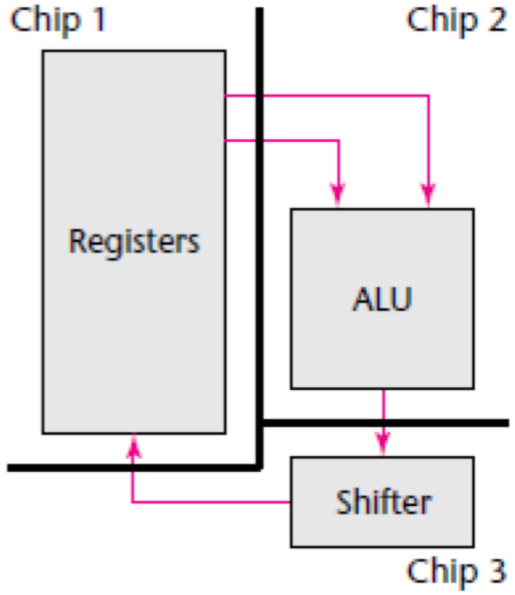
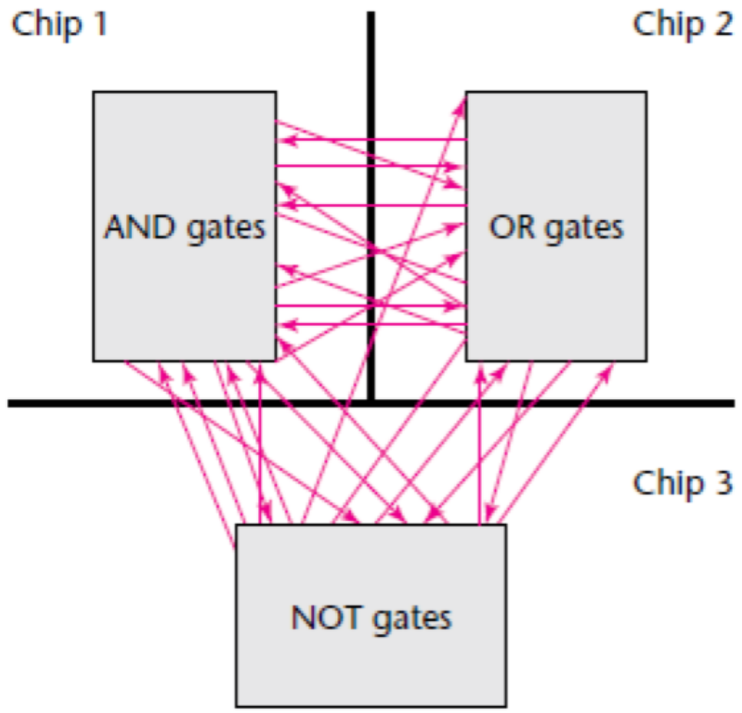


FIGURE 7.3 The computer of Figure 7.1 fabricated on three other chips.



好的分解和坏的分解

```
*#*#
```

//打印2对 * #

```
System.out.print(“*”);
System.out.print(“ # ”);
```


```
System.out.print(“*”);
System.out.print(“ # ”);
```

```
*#*#*#...
```

```
//打印100对 * #
```

我们愿意输入200遍么？

System.out.print(“*”);

System.out.print(“ # ”);

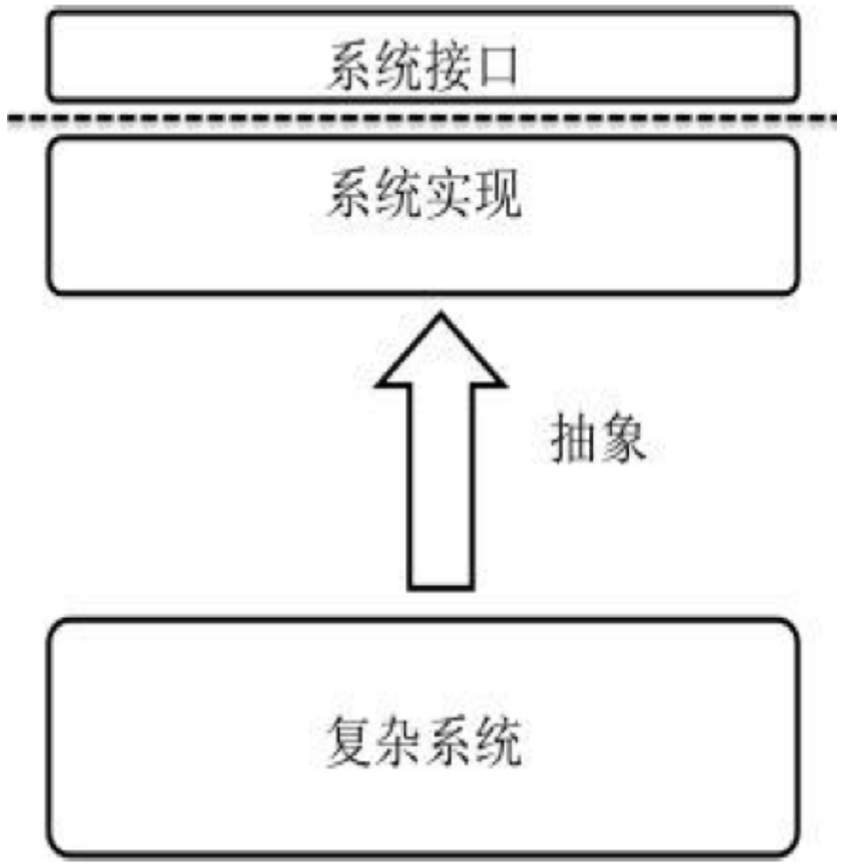
…

System.out.print(“*”);

System.out.print(“ # ”);

降低复杂度的方法 二

抽象



抽象

```
void printSingleStarSharp(){
    System.out.print(“*”);
    System.out.print(“ # ”);
}

void printStars(){
    printSingleStarSharp();
    printSingleStarSharp();
    …//100遍
}
```

抽象的关键点

- 抽象之后，接口的复杂度变小，
- 接口和实现之间达成一种契约。