

数据的机器级表示与运算

整数的除法

主讲人：邓倩妮

上海交通大学

部分内容来自：

1. 《深入理解计算机系统》第三版，机械工业出版社，作者：Bryant, R. E. 等
2. *Computer Organization and Design, 4th Edition*, Patterson & Hennessy



上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

定点数除法运算



手工计算过程:

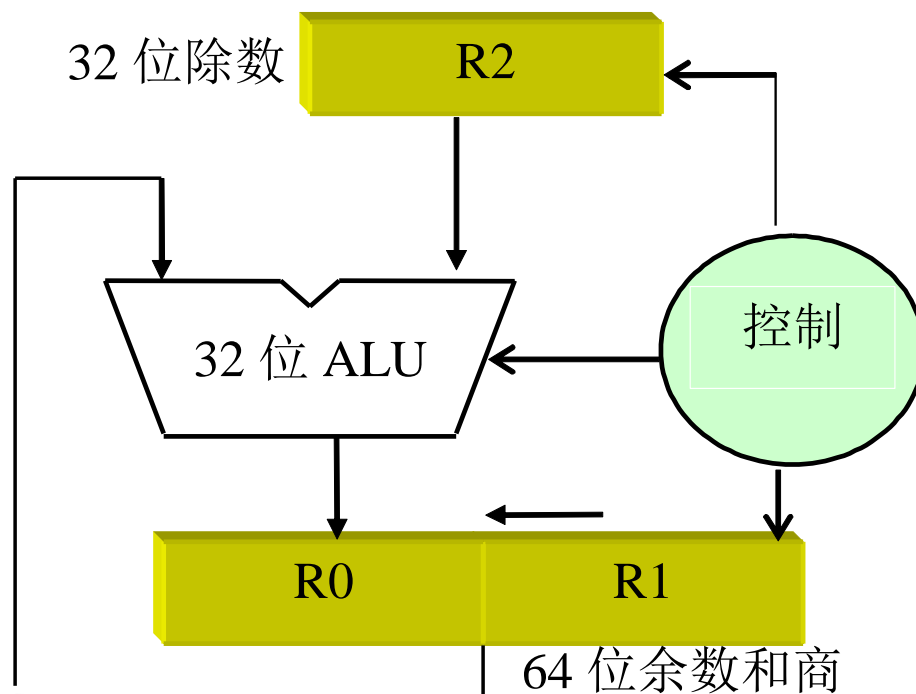
$$\begin{array}{r} \overline{0011} \\ 0010 \overline{) 00000111} \\ \underline{0010} \\ 00011 \\ \underline{0010} \\ 0001 \end{array}$$

硬件计算过程:



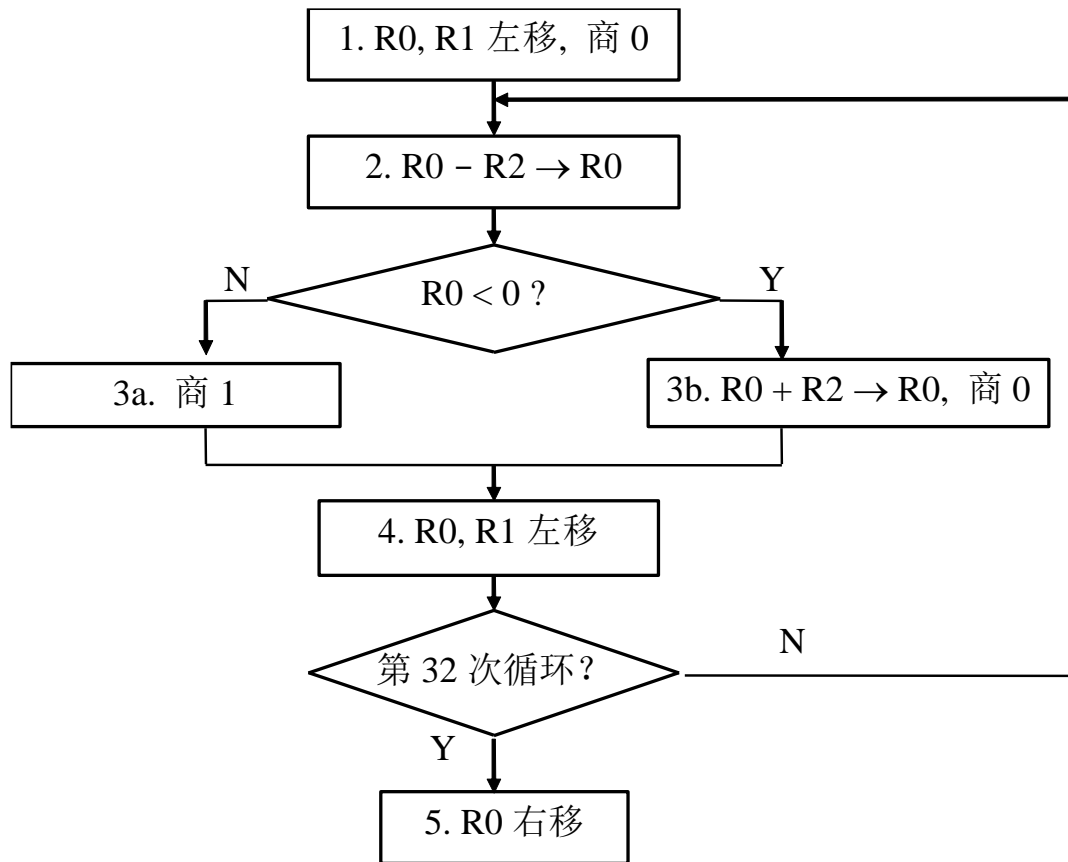
[illegible]

除法器电路



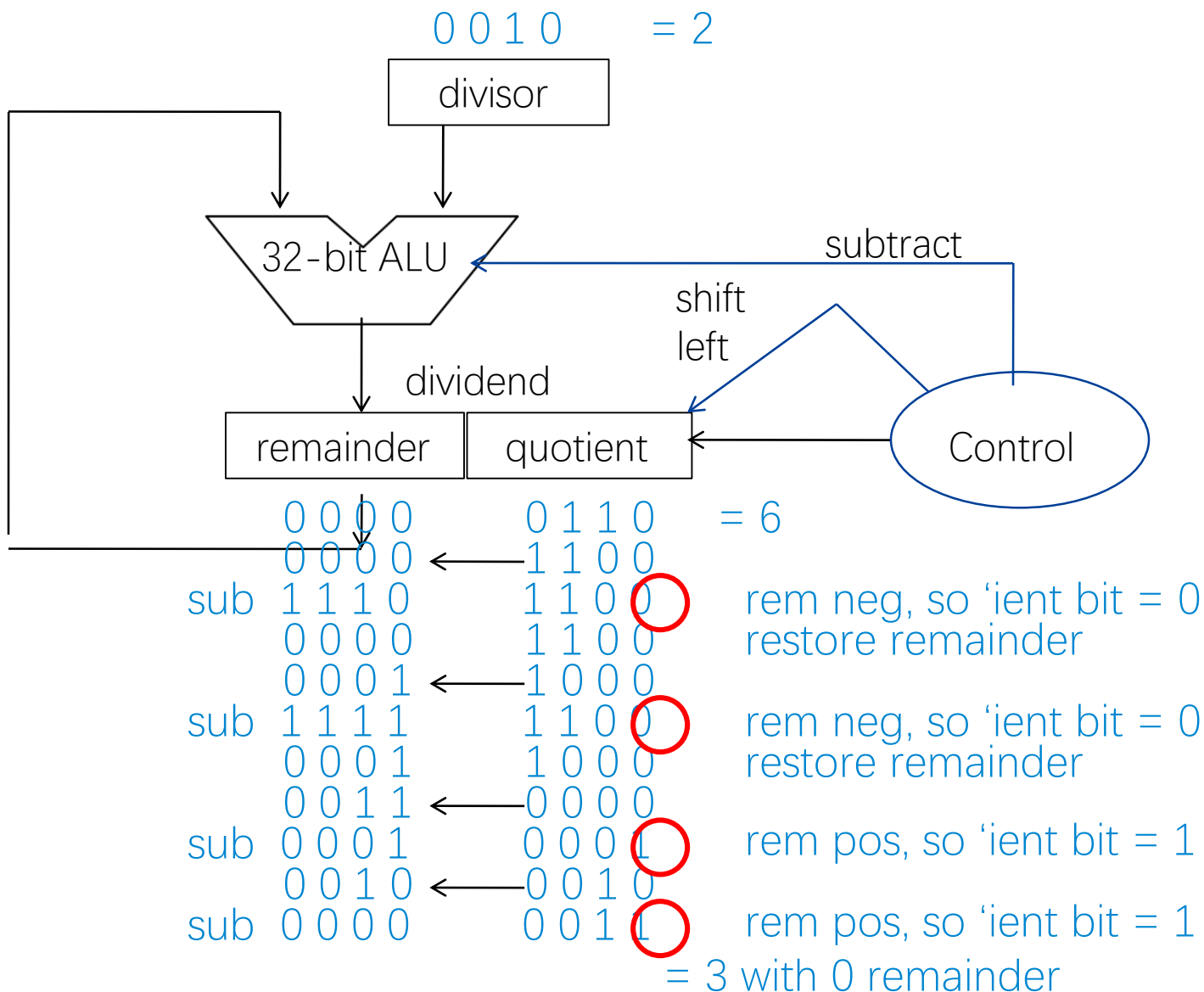
实现 32 位定点原码一位除法的方案

无符号除法流程



实现 32 位定点原码一位除法的流程图

无符号除法 Hardware 举例



有符号除法



- 被除数 = 商 * 除数 + 余数
- 为了保证：
- $-(x \div y) = (-x) \div y$
- 两者运算得到的商的绝对值相等
- 例如：
- $(+7) \div (-2)$ 商： -3； 余数： +1
- $(-7) \div (-2)$ 商： 3； 余数： -1
- 余数符号与被除数符号相同

MIPS 中的除法

- Divide (`div` and `divu`) generates the remainder in `hi` (余数) and the quotient in `lo` (商)

```
div    $s0, $s1          # lo = $s0 / $s1
                        # hi = $s0 mod $s1
```

0	16	17	0	0	0x1A
---	----	----	---	---	------

- Instructions `mfhi rd` and `mflo rd` are provided to move the quotient and remainder to (user accessible) registers in the register file
- ❑ 除法忽略溢出，需要软件自行检查商是否太大、以及避免除数为 0.

通过右移位实现除以 2 的幂



- $u \gg k$ 等价于 $u \div 2^k$ 吗 ?
- $u \gg k$ gives $\lfloor u / 2^k \rfloor$
- 对无符号数、逻辑右移

	Division	Computed	Hex	Binary
x	15213	15213	3B 6D	00111011 01101101
x >> 1	7606.5	7606	1D B6	00011101 10110110
x >> 4	950.8125	950	03 B6	00000011 10110110
x >> 8	59.4257813	59	00 3B	00000000 00111011

无符号除法（除以 2 的幂）的编译优化



```
unsigned long udiv8
(unsigned long x)
{
    return x/8;
}
```

C Function

Compiled Arithmetic Operations

```
shrq      $3, %rax
```

Explanation

```
# Logical shift
return x >> 3;
```

- 无符号数、正整数使用逻辑移位 (logical shift)
- Java : Logical shift written as >>>

通过右移位实现除以 2 的幂（续）



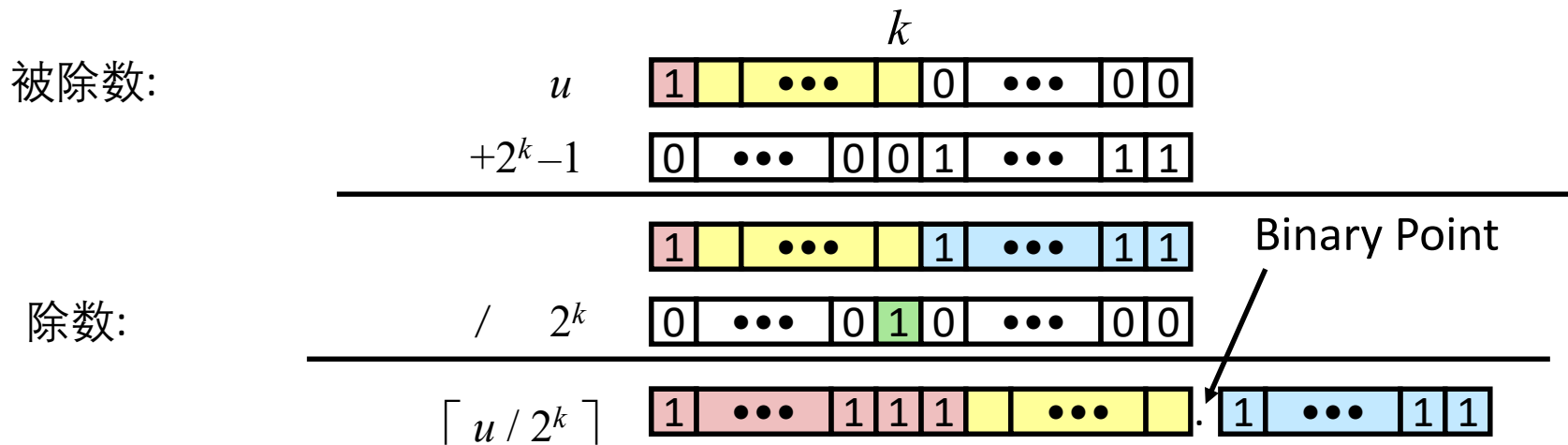
- $u \gg k$ 等价于 $u \text{ */ } 2^k$ 吗？
- $u \gg k$ gives $\lfloor u / 2^k \rfloor$
- 对有符号数、算术右移

	Division	Computed	Hex	Binary
y	-15213	-15213	C4 93	11000100 10010011
y >> 1	-7606.5	-7607	E2 49	11100010 01001001
y >> 4	-950.8125	-951	FC 49	11111100 01001001
y >> 8	-59.4257813	-60	FF C4	11111111 11000100

负数除以 2 的幂：修正

- Want $\lceil \mathbf{x} / 2^k \rceil$ (Round Toward 0 : 向0方向舍入)
- 表达为： $\lfloor (\mathbf{x} + 2^k - 1) / 2^k \rfloor$,
- In C: $(\mathbf{x} + (1 \ll k) - 1) \gg k$

Case 1: 不需要舍入 No rounding

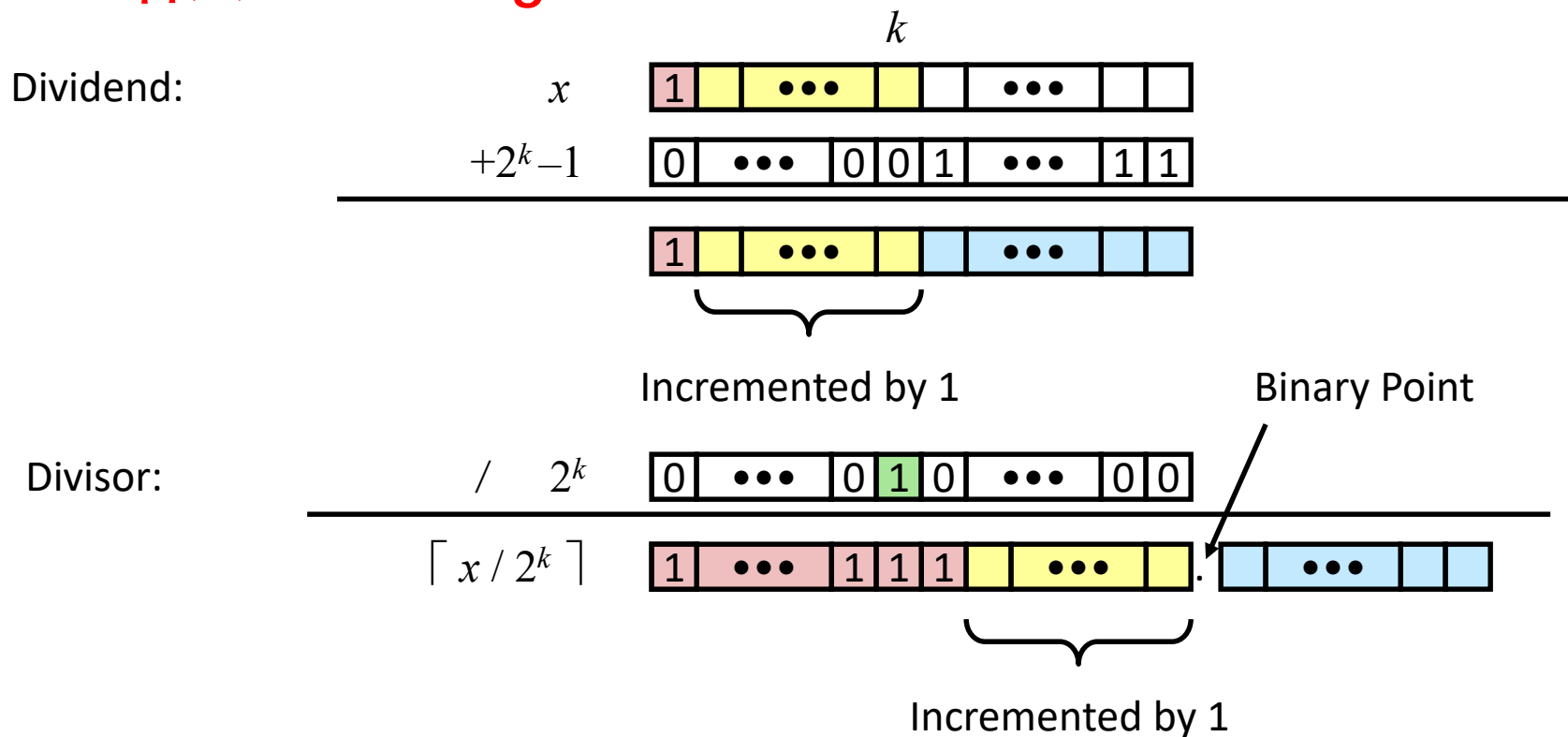


修正与不修正结果一样

负数除以 2 的幂：修正(Cont.)



Case 2: 舍入 Rounding



通过修正，向零方向舍入

有符号除法（除以 2 的幂）的编译优化



```
long idiv8(long x)
{
    return x/8;
}
```

C Function

Compiled Arithmetic Operations

```
testq %rax, %rax
js    L4
L3:
    sarq    $3, %rax
    ret
L4:
    addq    $7, %rax
    jmp    L3
```

Explanation

```
if x < 0
    x += 7;
# Arithmetic shift
return x >> 3;
```

- 算术移位 for int
- Java : Arith. shift written as >>

谢谢！

