Chap. 3

Arithmetic for Computers

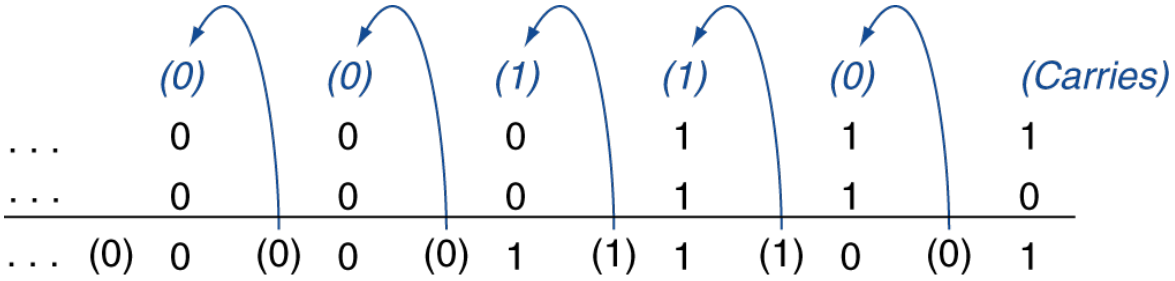
Arithmetic for Computers

- Operations on integers
  - Addition and subtraction
  - Multiplication and division
  - Dealing with overflow

- Floating-point (浮動小數點) real numbers
  - Representation and operations

Integer Addition

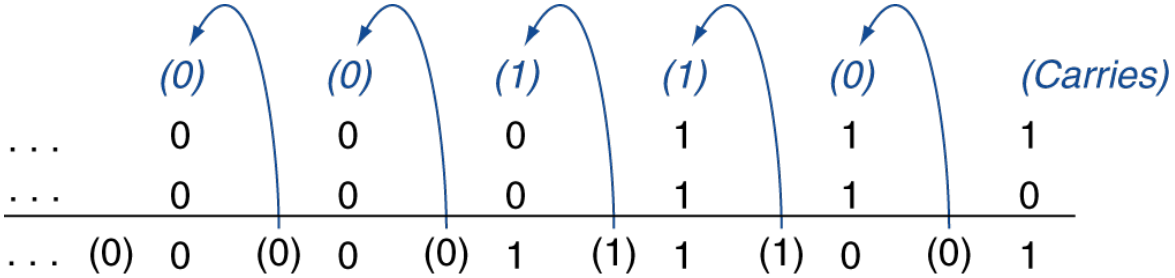
- Example : 7 + 6



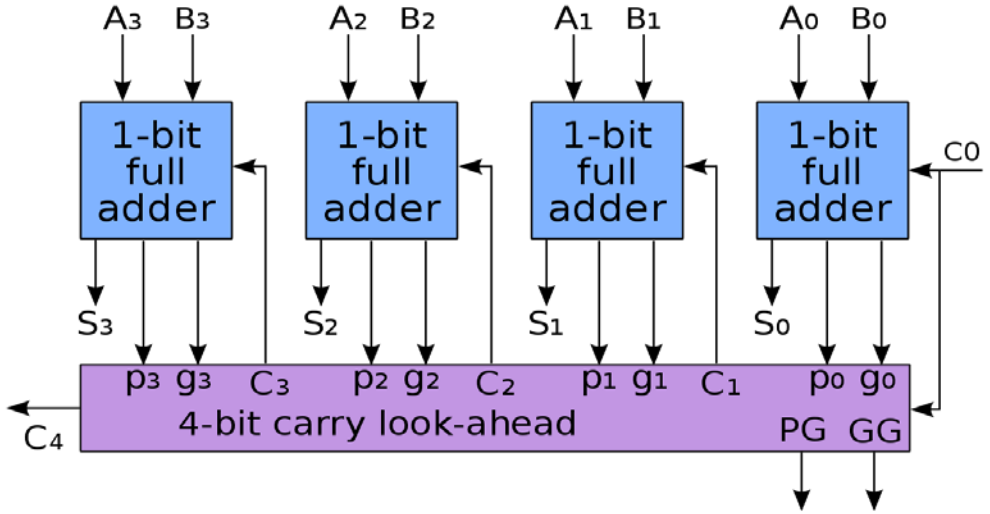
- Half Adder
- Full Adder
- Parallel Adder
- Carry Lookahead Adder

Integer Addition

- Example : 7 + 6



carry lookahead adder(CLA)

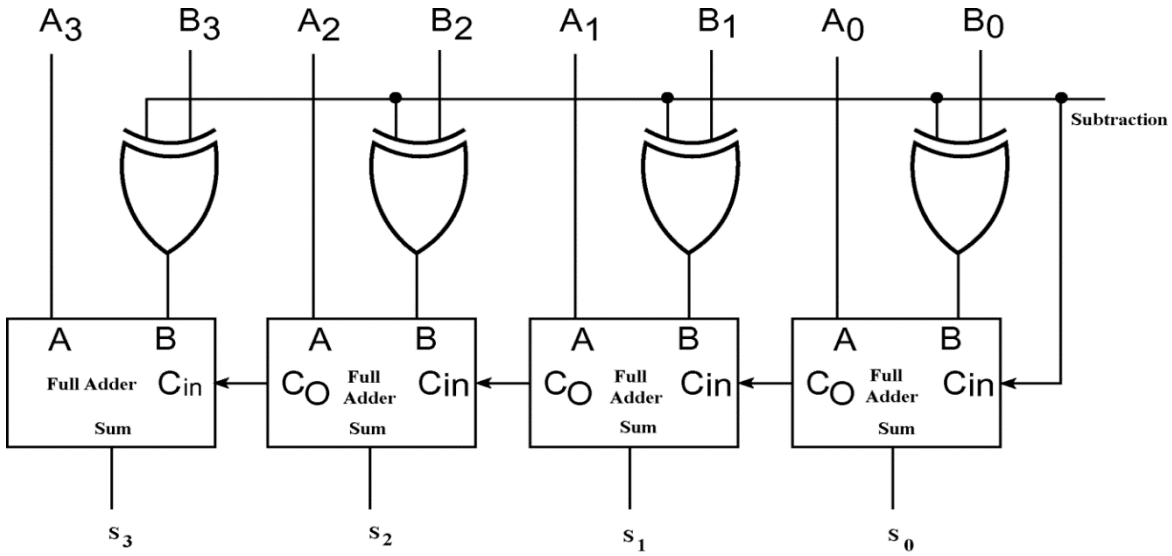


Integer Subtraction

- Add negation of second operand
- Example : 7 – 6 = 7 + (–6)

+7:	0000 0000 … 0000 0111
–6:	1111 1111 … 1111 1010
+1:	0000 0000 … 0000 0001

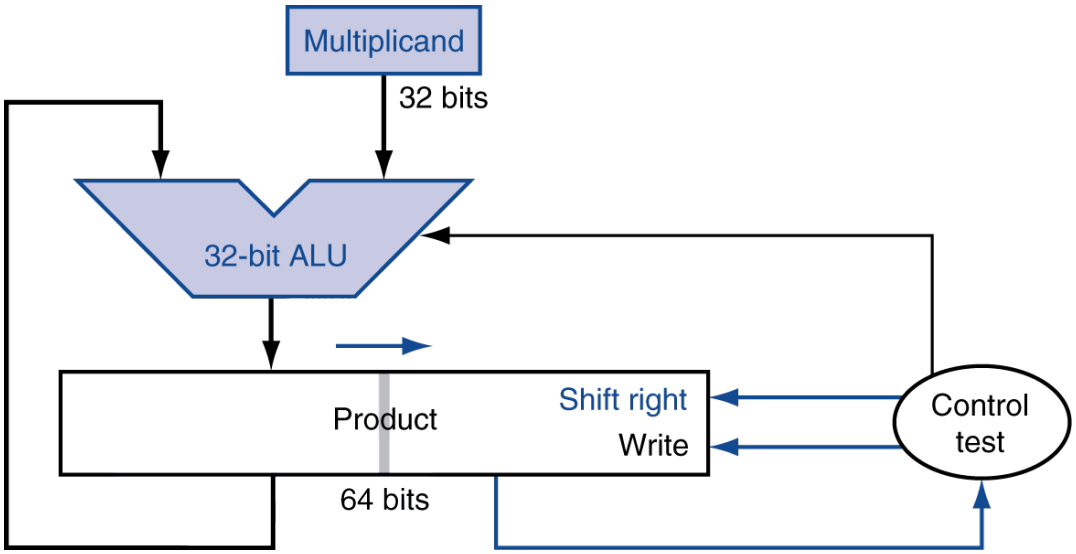
- Binary Adder/Subtractor



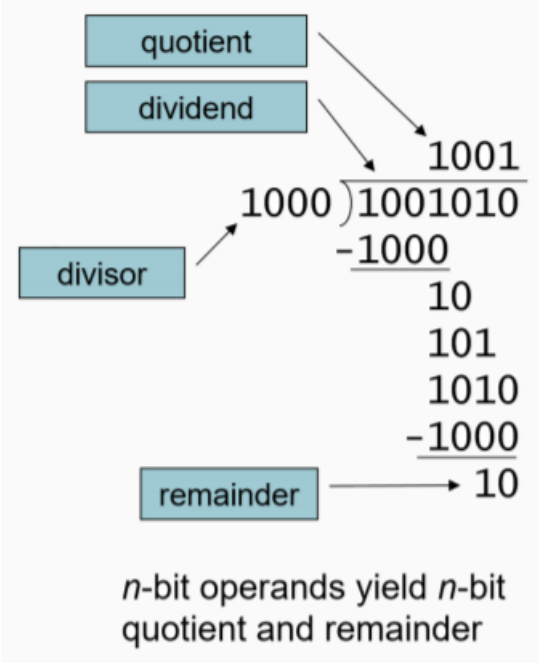
Optimized Multiplier (Fig.3.5)

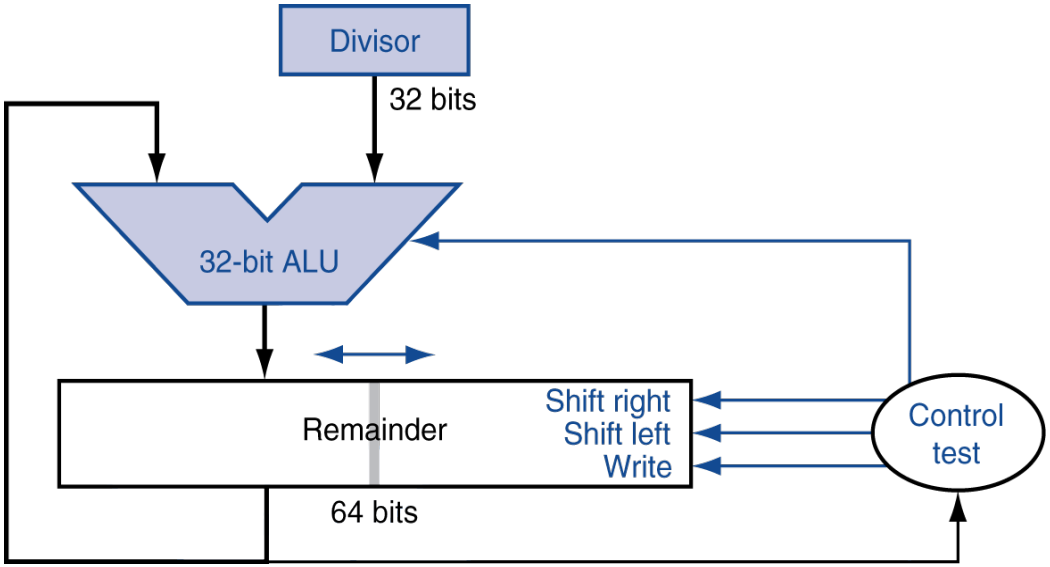
- Perform steps in parallel: add/shift

```
      1000
x     1001
=======
      1000
     0000
    0000
+1000
========
 1001000
```



Optimized Divider (Fig. 3.11)





- Looks a lot like a multiplier!
  - Same hardware can be used for both

Floating Point

- Representation for non-integral numbers
  - Including very small and very large numbers
- Like scientific notation
  - -2.34×10^{56} ← normalized
  - $+0.002 \times 10^{-4}$ ← not normalized
  - $+987.02 \times 10^{9}$ ← not normalized
- In binary
  - $\pm 1.xxxxxxx_2 \times 2^{yyyy}$
- Types `float` and `double` in C

Floating Point Standard

- Defined by IEEE Std 754-1985
- Developed in response to divergence of representations
  - Portability issues for scientific code
- Now almost universally adopted

- Two representations
  - Single precision (32-bit)
  - Double precision (64-bit)

  - Precision vs. range

IEEE Floating-Point Format

single: 8 bits
double: 11 bits

single: 23 bits
double: 52 bits

$$x = (-1)^S \times (1 + \text{Fraction}) \times 2^{(\text{Exponent} - \text{Bias})}$$

- S: sign bit (0 $\Rightarrow$ non-negative, 1 $\Rightarrow$ negative)
- Exponent: excess representation: actual exponent + Bias
  - Ensures exponent is unsigned
  - Single: Bias = 127; Double: Bias = 1203

Subword Parallellism

- Graphics and audio applications can take advantage of performing simultaneous operations on short vectors
  - Example: 128-bit adder:
    - Sixteen 8-bit adds
    - Eight 16-bit adds
    - Four 32-bit adds

- Also called data-level parallelism, vector parallelism, or
- Single Instruction, Multiple Data (SIMD)

x86 FP Architecture

- Originally based on **8087 FP coprocessor**
  - 8 × 80-bit extended-precision registers
  - Used as a push-down stack
  - Registers indexed from TOS: ST(0), ST(1), …
- Very difficult to generate and optimize code
  - Result : poor FP performance

Streaming SIMD Extension 2 (SSE2)

- Can be used for multiple FP operands
  - 2 × 64-bit double precision
  - 4 × 32-bit double precision
  - Single-Instruction Multiple-Data

Concluding Remarks

- ISAs support arithmetic
  - Signed and unsigned integers
  - Floating-point approximation to reals
- Bounded range and precision
  - Operations can overflow and underflow

- MIPS ISA
  - Core instructions : 54 most frequently used
    - 100% of SPECINT, 97% of SPECFP
  - Other instructions: less frequent