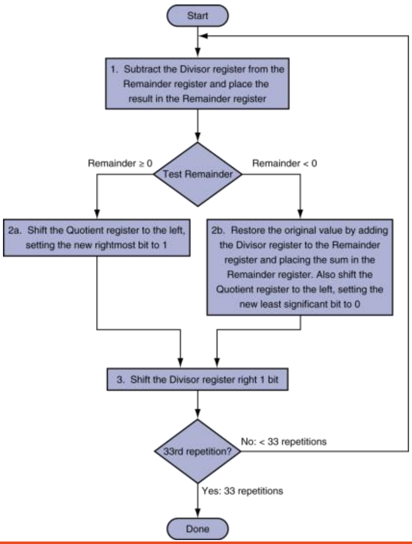
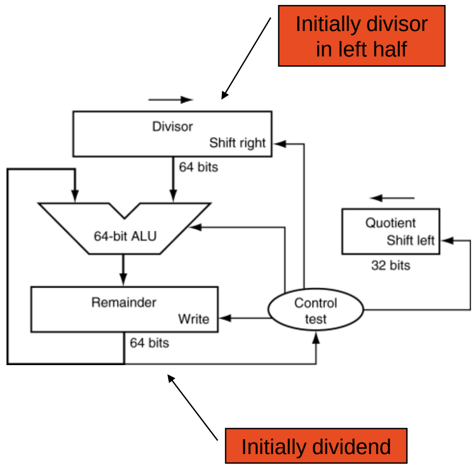
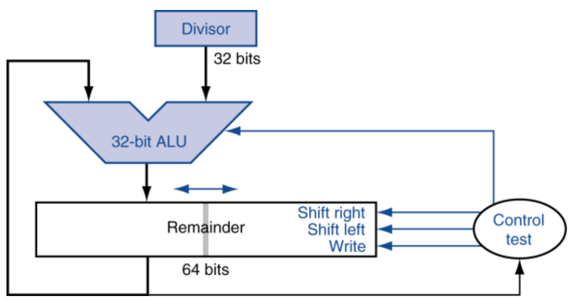
Division Hardware





Optimized Divider



- One cycle per partial-remainder subtraction
- Looks a lot like a multiplier!
  - Same hardware can be used for both

MIPS Division

- Use HI/LO registers for result
  - HI: 32-bit remainder
  - LO: 32-bit quotient
- Instructions
  - `div rs, rt / divu rs, rt`
  - No overflow or divide-by-0 checking
    - Software must perform checks if required
  - Use `mfhi`, `mflo` to access result

CSULB

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 → $+2.0 \times 10^{-7}$
  - $+987.02 \times 10^{9}$ ← not normalized → $+9.8702 \times 10^{7}$
- In binary
  - $\pm 1.xxxxxxx_2 \times 2^{yyyy}$
- Types `float` and `double` in C

CSULB

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)

CSULB

IEEE Floating-Point Format

	single: 8 bits double: 11 bits	single: 23 bits double: 52 bits
S	Exponent	Fraction

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

- S: sign bit (0 ⇒ non-negative, 1 ⇒ negative)
- Normalize significand: 1.0 ≤ |significand| < 2.0
  - Always has a leading pre-binary-point 1 bit, so no need to represent it explicitly (hidden bit)
  - Significand is Fraction with the "1." restored
- Exponent: excess representation: actual exponent + Bias
  - Ensures exponent is unsigned
  - Single: Bias = 127; Double: Bias = ~~1203~~ 1023

CSULB

Single-Precision Range

- Exponents 00000000 and 11111111 reserved
- Smallest absolute value
  - Exponent:
    $\Rightarrow$ actual exponent = 00 ~ 1 ⇒ −126
  - Fraction: 000…00 $\Rightarrow$ significand =
    $\pm 1.0 \times 2^{-126}$
- Largest absolute value
  - Exponent: 1111 ~ 0
    $\Rightarrow$ actual exponent = 254 − 127 = 127
  - Fraction: 111…11 $\Rightarrow$ significand ≈
    $\pm 1.999\ldots \times 2^{127}$

$1 \times 2^{1-127}$

S	Exponent	Fraction

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

Double-Precision Range

- Exponents 0000…00 and 1111…11 reserved
- Smallest value
  - Exponent: 00000000001
    $\Rightarrow$ actual exponent = 1 – 1023 = –1022
  - Fraction: 000…00 $\Rightarrow$ significand = 1.0
  - $\pm 1.0 \times 2^{-1022} \approx \pm 2.2 \times 10^{-308}$
- Largest value
  - Exponent: 11111111110
    $\Rightarrow$ actual exponent = 2046 – 1023 = +1023
  - Fraction: 111…11 $\Rightarrow$ significand ≈ 2.0
  - $\pm 2.0 \times 2^{+1023} \approx \pm 1.8 \times 10^{+308}$

S	Exponent	Fraction

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

Denormal Numbers

- Exponent = 000...0 $\Rightarrow$ hidden bit is 0
  - Denormal with fraction = 000...0

$$x = (-1)^S \times (0 + 0) \times 2^{-\text{Bias}} = \pm 0.0$$

Two representations of 0.0!

Sign	Exponent (*e*)	Fraction (*f*)	Value
0	00⋯00	00⋯00	+0
1	00⋯00	00⋯00	−0

CSULB

Denormal Numbers

- Exponent = 000...0 $\Rightarrow$ hidden bit is 0

$$x = (-1)^S \times (0 + \text{Fraction}) \times 2^{-\text{Bias}}$$

- Smaller than normal numbers
  - allow for gradual underflow, with diminishing precision

$0.\underbrace{0000\cdots1}_{23\text{ bits}} \times 2^{-127}$

Sign	Exponent (*e*)	Fraction (*f*)	Value
0	00⋯00	00⋯01 ⋮ 11⋯11	Positive Denormalized Real $0.f \times 2^{(-b+1)}$
1	00⋯00	00⋯01 ⋮ 11⋯11	Negative Denormalized Real $-0.f \times 2^{(-b+1)}$

CSULB

Infinities and NaNs

- Exponent = 111…1, Fraction = 000…0
  - ±Infinity
  - Can be used in subsequent calculations, avoiding need for overflow check
- Exponent = 111…1, Fraction ≠ 000…0
  - Not-a-Number (NaN)
  - Indicates illegal or undefined result
    - e.g., 0.0 / 0.0, 0.0 * ∞
  - Can be used in subsequent calculations

CSULB

Floating-Point Precision

$1,9999 \times 10^3$

$= 1999.\cancel{9} \Rightarrow 3$ digits

- Relative precision
  - all fraction bits are significant
  - Single: approx 2^{23}
    - Equivalent to $\log 2^{23} = 23 \cdot \log 2 \approx 6$ decimal digits of precision
  - Double: approx 2^{52}
    - Equivalent to $\log 2^{52} \approx 16$ decimal digits of precision

CSULB

Floating-Point Example

- What number is represented by the single-precision float

  1 10000001 01000…00

  $1.\underbrace{010000\sim0}_{23}$

  - S = 1
  - Fraction = $010\sim0 = 0100\sim0_2$
  - Exponent = $10000001 = 129$
- x = $(-1)^1 \times (1 + 01_2) \times 2^{129-127}$

  $= (-1) \times 1.25 \times 2^2$

$1.01_2 = 1.25_{10}$

$+1\times 2^0 + 0\times 2^{-1} + 1\times 2^{-2}$

Floating-Point Example

- Represent –0.75
  - –0.75 = $-0.5 - 0.25 = -0.11_2 = -1.1 \times 2^{-1}$
  - S = 1
  - Fraction = $10\sim0$
  - Exponent = $x - 127 = -1,\ x = 126$
    - Single:
    - Double:
- Single:
- Double: 0

//9/28