# 3. Sequential Algorithms for multiplication and division

fast multiplier : 6장
fast divider: 7장

## – Sequential multiplier

- multiplier X = $x_{n-1}\, x_{n-2} \ldots x_0$

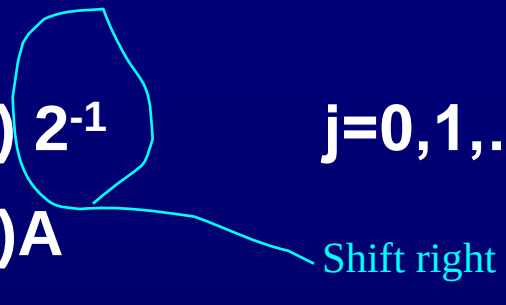
  multiplicand A = $a_{n-1}\, a_{n-2} \ldots a_0$

- sequential algorithm for multiplication

$$p^{(0)} = 0$$

$$p^{(j+1)} = (p^{(j)}_{k-1} + x_j A)\, 2^{-1} \qquad j=0,1,\ldots,n-2$$

$$p^{(n-1)} = \left(\sum_{i=0} x_i\, 2^{i-n+1}\right) A$$




- 두 operand가 모두 양수일 때 ($x_{n-1} = a_{n-1} = 0$)

  곱 $U = 2^{n-1}\, p^{(n-1)} = XA$

- 곱의 크기는 (2n-2) bits, 부호를 포함하여 (2n-1) bits
- signed-magnitude 수인 경우 : 부호bit 별도 계산

- **2's complement나 1's complement인 경우**
  **multiplicand A가 음수인 경우**
  **multiplier X가 음수인 경우** } 구분
- **multiplicand A만 음수인 경우 앞  알고리듬과 동일**

| | | |
|---|---|---|
| A | 1011 | -5 |
| X | x 0011 | x 3 |
| $p^{(0)}$ | 0000 | |
| $x_0$=1 add A | 1011 | |
| | 1011 | |
| shift | 11011 | |
| $x_1$=1 add A | 1011 | |
| | 110001 | |
| shift | 110001 | |
| $x_2$=0 shift | 1110001 | |
| $x_3$ sign bit: nop | | |
| shift left 3 bits | 1110001 | -15 |

- **X < 0 일 때 (2's complement)**

  $X = -x_{n-1}2^{n-1} + X$, where $X = \sum_{i=0}^{n-2} x_i 2^i$

  **multiplier의 sign bit를 제외하면**

  $U = XA = (X + x_{n-1}2^{n-1})A$

  $= XA + A\,x_{n-1}2^{n-1}$

  $\therefore\ XA = U - A\,x_{n-1}2^{n-1}$

Correction step is required

| | | |
|---|---|---|
| A | 1011 | -5 |
| X | x 1101 | x -3 |
| $x_0$=1 add A | 1101 | |
| shift | 11011 | |
| $x_1$=0 shift | 111011 | |
| $x_2$=1 add A | 1011 | |
| | 100111 | |
| shift | 1100111 | |
| $x_3$=1 correct step:add -A | 0101 | |
| | 0001111 | |
| shift left 3 bits | 0001111 | 15 |

- **X < 0 일 때 (1's complement)**

  $X = -x_{n-1}(2^{n-1}-1) + X$, where $X = \sum_{i=0}^{n-2} x_i 2^i$

$\therefore$ $XA = U - A\,x_{n-1}2^{n-1} + x_{n-1}\,A$

| | | | |
|---|---|---|---|
| A | | 0101 | 5 |
| X | x | 1100 | x -3 |
| $x_3$=1 $p^{(0)}$ = A | | 0101 | |
| $x_0$=0 shift | | 00101 | |
| $x_1$=0 shift | | 000101 | |
| $x_2$=1 add A | | 0101 | |
| | | 011001 | |
| shift | | 0011001 | |
| $x_3$=1 correct step:add A | | 1011111 | 1's complement extension |
| | | 1110000 | |
| shift left 3 bits | | 1110000 | -15 |

- **Sequential division**
  - 복잡하고 느리다
  - **dividend X, divisor D, quotient Q, remainder R**

    **X = Q D + R with R < D**
  - 우선 **X,D,Q,R** 모두 양수 가정

    **X: 2n-bit register**

    **D,Q,R: n-bit register**
  - **Q < $2^{n-1}$** 이기 위하여 **X< $2^{n-1}$D**

    이 조건을 만족하지 않으면 **preshifting**이 필요
  - **D=0 ⇒ divide by zero flag**

- X,D,Q,R 모두 fraction으로 해석할 때 알고리듬이 단순함

  X < D : overflow condition

  $Q = 0.q_1 \ldots q_m$

- repeat subtractions and shifts
- $q_i$ 는 나머지 $2\,q_{i-1}$ 와 D를 비교하여 결정

  $r_i = 2r_{i-1} - q_i D \qquad (i=1,2,\ldots,m)$

  $r_0 = 0$

  $X = QD + r_m\, 2^{-m}$

  $R = r_m\, 2^{-m}$

어려운 과정

- **X = 0.100000 =1/2**
  **D = 0.110 =3/4** **X<D 만족**

Extra bit is required

$2r_i$와 D를 비교하여 결정

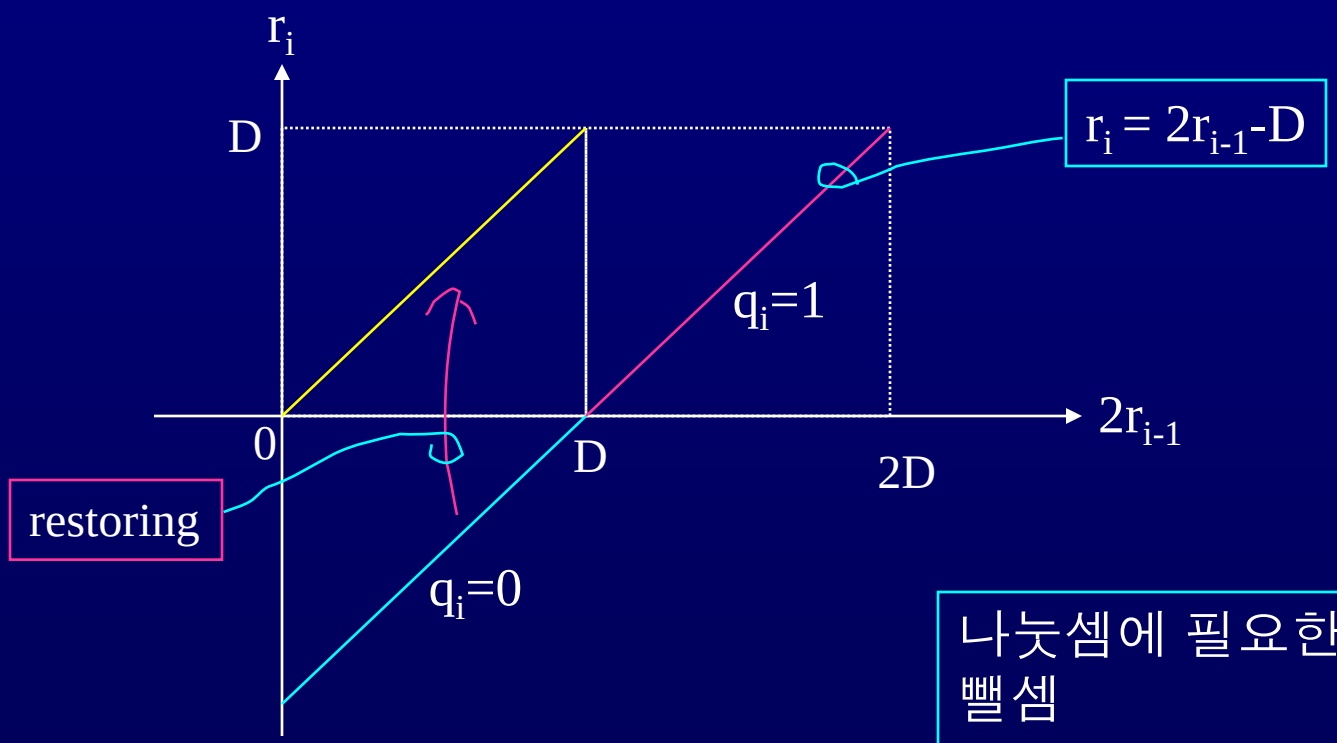
| | | |
|---|---|---|
| $r_0$=X | 0.100 000 | |
| 2 $r_0$ | 01.000 00 0 | set $q_1$=1 |
| add-D | 11.010 | |
| $r_1$=2 $r_0$ -D | 00.010 000 | |
| $2r_1$ | 00.100 000 | set $q_2$=0 |
| $r_2$=2 $r_1$ | 01.000 000 | |
| $2r_2$ | 10.000 000 | set $q_3$=1 |
| add-D | 11.010 | |
| $r_3$=2 $r_2$ -D | 00.010 000 | |

Q= 0.101 = 5/8

R= $r_m 2^{-m}$ = 1/4 x 2 = 1/32

- **Restoring division**
  - **Quotient bit 결정**

    **$2r_{i-1}$ - D 를 한 후 그의 부호로 결정**
  - **양이면 $q_i$ =1**
  - **음이면 $q_i$ =0, 그리고 다시 D를 더해 주어야 한다.**




나눗셈에 필요한 연산

뺄셈 : m 개

shift : m 개

덧셈 또는 copy 보관 : m/2 개

- **Nonrestoring division**
  - 나머지가 음수가 되더라도 나머지를 원래 값으로 복원시키는 작업을 따로 하지 않고 다음 나머지 계산하는 step에서 반영한다.
  - $2r_{i-1} - D < 0$일때 다음 나머지 계산하는 step에서

$2(2r_{i-1} - D)+D = 4r_{i-1} - D$ (in nonrestoring division )

shift하고 D를 더한다. 결과 같음

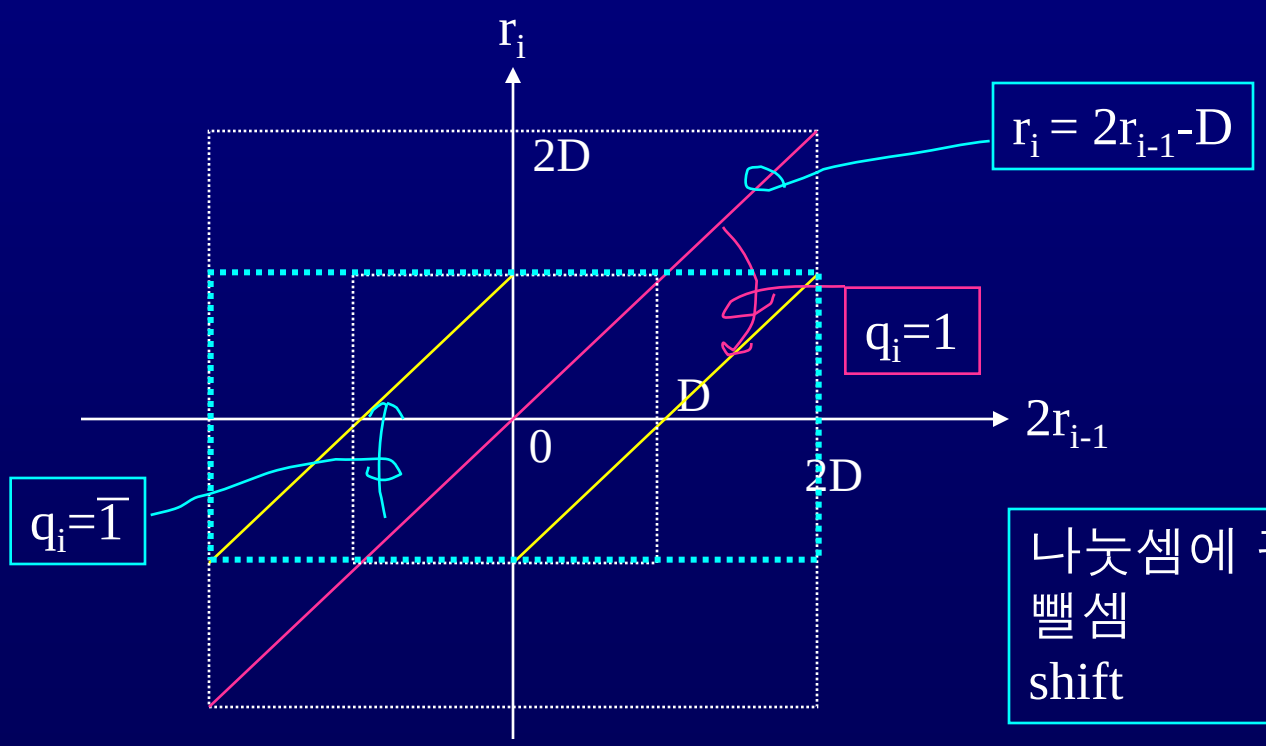
$2[(2r_{i-1} - D)+D] - D = 4r_{i-1} - D$ (in restoring division )

restoring하고 shift하고 D를 뺀다.

- **Nonrestoring division**
  - **quotient bit를 잘못 선택한 것을 보상할 수 있도록 음수의 quotient bit를 허용한다.**

$$q_i = \begin{cases} 1 & \text{if } 2r_{i-1} \geq 0 \\ \bar{1} & \text{if } 2r_{i-1} < 0 \end{cases}$$

주: $q_i \neq 0$




나눗셈에 필요한 연산

| | |
|---|---|
| 뺄셈 | : m 개 |
| shift | : m 개 |

- **X = 0.100000 =1/2**
  **D = 0.110 =3/4**

| | | |
|---|---|---|
| $r_0$=X | 0.100 000 | |
| 2 $r_0$ | 01.000 00 0 | set $q_1$=1 |
| add -D | 11.010 | |
| $r_1$ | 00.010 000 | |
| $2r_1$ | 00.100 000 | set $q_2$=1 |
| add -D | 11.010 000 | |
| $r_2$ | 11 110 000 | |
| $2r_2$ | 11 100 000 | set $q_3=\bar{1}$ |
| add D | 00.110 000 | |
| $r_3$ | 00.010 000 | |

$Q = 0.11\bar{1} = 5/8$

$R = r_m 2^{-m} = 1/4 \times 2 = 1/32$

- **Nonrestoring division 장점**
  - **2's complement number로 확장 용이**
  - **$q_i$ = 1 if $2r_{i-1}$ and D have the same sign**
    **$\bar{1}$ if $2r_{i-1}$ and D have opposite signs**

- **X = 0.100000 =1/2**
  **D = 1.010 = -3/4**

| | | |
|---|---|---|
| $r_0$=X | 0.100 000 | |
| 2 $r_0$ | 01.000 00 0 | set $q_1=\bar{1}$ |
| add D | 11.010 | |
| $r_1$ | 00.010 000 | |
| $2r_1$ | 00.100 000 | set $q_2=\bar{1}$ |
| add D | 11.010 000 | |
| $r_2$ | 11 110 000 | |
| $2r_2$ | 11 100 000 | set $q_3$=1 |
| add -D | 00.110 000 | |
| $r_3$ | 00.010 000 | |

Q= $0.\bar{1}\bar{1}1$ = $0.\bar{1}0\bar{1}$ =- 5/8

R= $r_m 2^{-m}$ = 1/4 x 2 = 1/32

- 마지막 나머지는 **dividend**와 같은 부호를 가져야 한다. **(정의)**
  - **X = 0.101000 =5/8**
    **D = 0.110 =3/4**

| | | |
|---|---|---|
| $r_0$=X | 0.101 000 | |
| 2 $r_0$ | 01.010 00 0 | set $q_1$=1 |
| add -D | 11.010 | |
| $r_1$ | 00.100 000 | |
| 2$r_1$ | 01.000 000 | set $q_2$=1 |
| add -D | 11.010 000 | |
| $r_2$ | 00.010 000 | |
| 2$r_2$ | 00.100 000 | set $q_3$=1 |
| add -D | 11.010 000 | |
| $r_3$ | 11.110 000 | X와 부호가 다르다 |
| add D | 00.110 000 | |
| $r_{3c}$ | 00.100 000 | |

Q= 0.111

$Q_c$ =Q- ulp = 0.110 = 3/4

R= $r_m 2^{-m}$ = 1/2 x $2^{-3}$ = 1/16

◆ 마지막 나머지 $r_{mc}$ 는 dividend X와 다른 부호를 가지면 correction 필요

1. Dividend X와 divisor D가 같은 부호인 경우

$$r_{mc} = r_m + D$$
$$Q_c = Q - ulp$$

2. Dividend X와 divisor D가 다른 부호인 경우

$$r_{mc} = r_m - D$$
$$Q_c = Q + ulp$$

붉은 색 부분은 $r_m$ 와 부호가 다르고 절대값은 크다.
∴ $r_{mc}$ 은 $r_m$ 와 부호가 다르다.
∴ $r_{mc}$ 은 X 와 부호가 같다.

- **X = 1.101000 =-3/8**
  **D = 0.110 =3/4**

답: Q=-1/2, 나머지 0

| | | |
|---|---|---|
| $r_0$=X | 1.101 000 | |
| 2 $r_0$ | 11.010 00 0 | set $q_1=\bar{1}$ |
| add D | 00.110 | |
| $r_1$ | 00.000 000 | zero remainder |
| $2r_1$ | 00.000 000 | set $q_2$=1 |
| add -D | 11.010 000 | |
| $r_2$ | 11.010 000 | |
| $2r_2$ | 10.100 000 | set $q_3=\bar{1}$ |
| add D | 00.110 000 | |
| $r_3$ | 11.010 000 | |
| add D | 00.110 000 | |
| $r_{3c}$ | 00.000 000 | |

$Q = 0.\bar{1}11\bar{1} = -3/8$

$Q_c = Q - ulp = 0.\bar{1}10 = -1/2$

중간 나머지가 zero인 것을 detect하면
$r_m$ 과 X 가 같은 부호이더라도 correction 필요
(to obtain zero remainder)

- 나눗셈 에서 **2's complement** 몫을 얻는 방법

$Q = 0.q_1 \ldots q_m, \qquad q_i \in \{1, \bar{1}\}$

아래 mapping으로

$q_i = \bar{1} \rightarrow p_i = 0$
$q_i = 1 \rightarrow p_i = 1$

$p_i = 1/2\,(1 + q_i)$

Q에서 P를 만든다

$P = 0.p_1 \ldots p_m, \qquad p_i \in \{0,1\}$

그리고,

1. Shift left 1 bit
2. Complement MSB
3. Shift 1 into LSB

$Q_2 = (1- p_1).p_2 \ldots p_m 1$

나눗셈 과정 중에서 bit serial하게 2's complement를 구할 수 있음