

# COL216

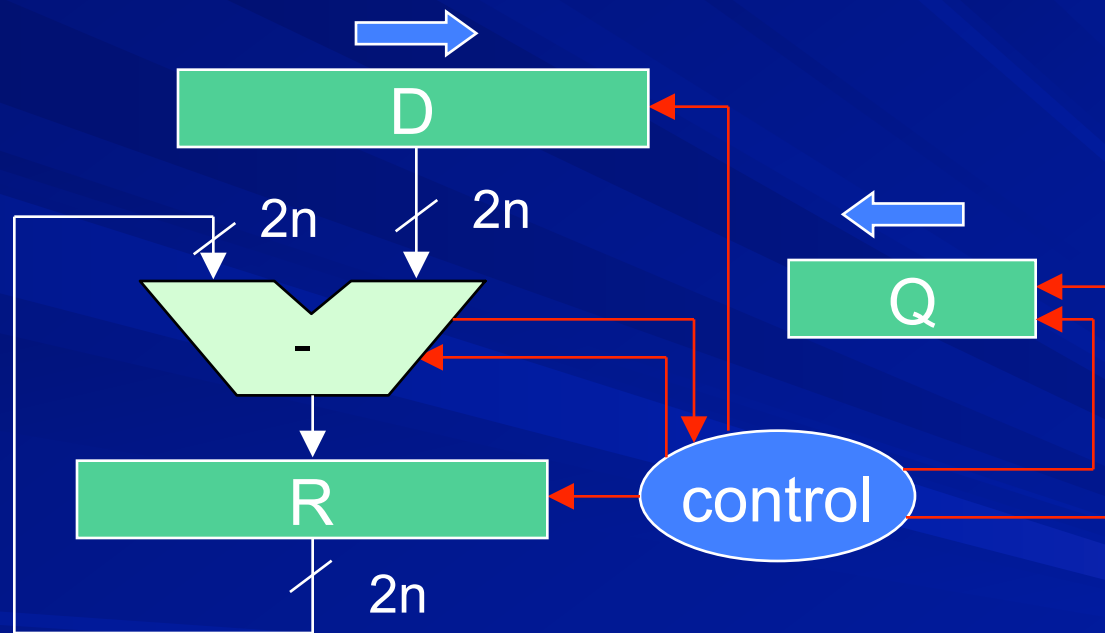
## Computer Architecture

Divider Design continued  
15th February, 2016

# Division algorithm

```
i = 0; Q = 0;  
R = A;  
D = B x 2n-1  
do {  
    if (D ≤ R)  
        R = R - D;  
        Q = 2 x Q + 1  
    else  
        Q = 2 x Q  
    D = D / 2; i ++  
} while (i < n)
```

# Divider design - 1



# Division algorithm

```
Q = 0; R = A;  
i = 0; D = B x 2n-1  
do {  
    if (D ≤ R)  
        R = R - D;  
        Q = 2 x Q + 1  
    else  
        Q = 2 x Q  
  
    D = D / 2; i ++  
} while (i < n)
```

```
Q = 0; R = A;  
i = 0; D = B x 2n-1  
do {  
    R = R - D;  
    if (0 ≤ R)  
        Q = 2 x  
    else  
        R = R + D;  
        Q = 2 x Q  
  
    D = D / 2; i ++  
} while (i < n)
```

Restoring  
step

# Delaying restoration

Restoring now ( $R = R + D$ ) and initial subtraction of the next iteration ( $R = R - D / 2$ ) is equivalent to

Doing initial addition (in place of subtraction) in the next iteration ( $R = R + D / 2$ )

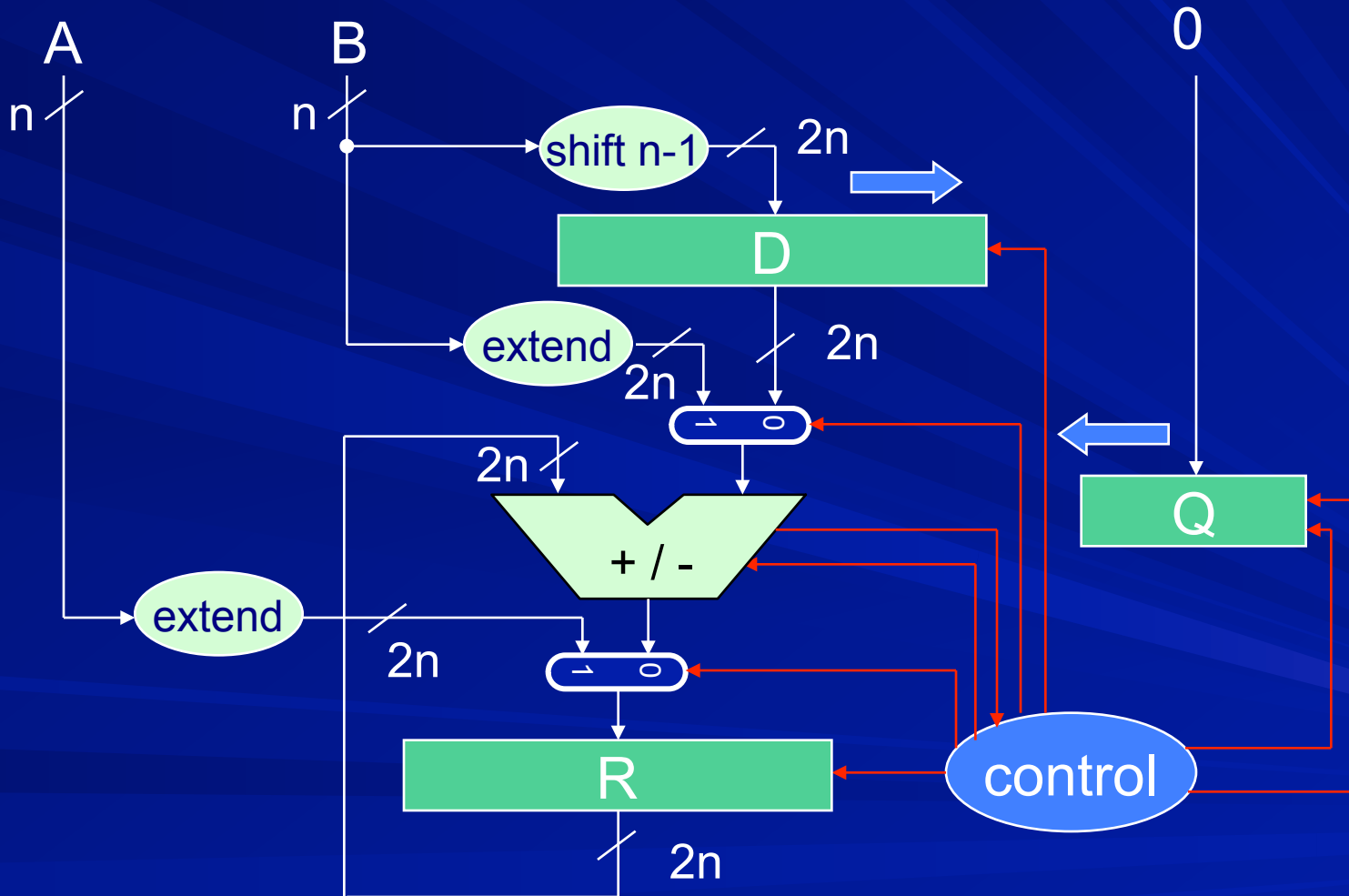
# Improved division algorithm

```
step1: i = 0; R = A; Q = 0; D = B x 2n-1
do {
  step2A:
    if (R < 0) R = R + D else R = R - D
  step2B:
    if (R < 0) Q = 2 x Q else Q = 2 x Q + 1
    D = D / 2; i ++
} while (i < n)
if (R < 0) R = R + B
```

# Division: example

$$\begin{array}{r}
 \begin{array}{c} 0100 \\ \uparrow \\ B \end{array} \bigg| \begin{array}{r} 00001101 \\ -0100 \\ \hline 11101101 \\ +0100 \\ \hline 11111101 \\ +0100 \\ \hline 00000101 \\ -0100 \\ \hline 00000001 \end{array}
 \end{array}
 \begin{array}{l}
 \leftarrow Q \\
 \leftarrow A \\
 \leftarrow -B \times 2^3 \\
 \leftarrow +B \times 2^2 \\
 \leftarrow +B \times 2^1 \\
 \leftarrow -B \times 2^0 \\
 \leftarrow R
 \end{array}$$

# Divider design





# Improving further

```
step1: i = 0; R = A; Q = 0; D = B x 2n-1
```

```
do {
```

```
  step2:
```

```
    if (R < 0)
```

```
      R = R + D; Q = 2 x Q - 1
```

```
    else
```

```
      R = R - D; Q = 2 x Q + 1
```

```
      D = D / 2; i ++
```

```
} while (i < n)
```

```
if (R < 0) R = R + B; Q = Q - 1
```

# Extending to signed division

step1:  $i = 0$ ;  $R = A$ ;  $Q = 0$ ;  $D = B \times 2^{n-1}$

do {

  step2:

    if ( $R_{n-1} \neq D_{n-1}$ )  $R = R + D$ ;  $Q = 2 \times Q - 1$

    else  $R = R - D$ ;  $Q = 2 \times Q + 1$

$D = D / 2$ ;  $i++$

} while ( $i < n$ )

if ( $R_{n-1} \neq A_{n-1}$ )

  if ( $R_{n-1} \neq D_{n-1}$ )  $R = R + B$ ;  $Q = Q - 1$

  else  $R = R - B$ ;  $Q = Q + 1$

# Signs in signed division

Dividend	Divisor	Quotient	Remainder
+	+	+	+
+	-	-	+
-	+	-	-
-	-	+	-