Toán học số 16 bit

TS Nguyễn Hồng Quang



Electrical Engineering

1

Cộng hai số 16 bít

	100's	10's	1's
	1	5	6
+	2	4	8
=	4	0	4

	256's	1's
	1A	44
+	22	DB
=	3D	1F

- 65535 + 65535 = 131070 số 17bit
- 44 + DB = 11F
- 1A + 22 = 3D



Electrical Engineering

Thuật toán

	65536' s	256's	1's
		R6	R7
+		R4	R5
=	R1	R2	R3

- Số thứ nhất: Byte cao
 R6 byte thấp là R7
- Số thứ 2: byte cao là R4 và byte thấp là R5)
- Kết quả lưu trong R1, R2, và R3.
- R3 chỉ lưu trữ một bit



Electrical Engineering

3

Các bước chương trình

MOV A,R7 ;Move the low-byte into the accumulator
ADD A,R5 ;Add the second low-byte to the accumulator
MOV R3,A ;Move the answer to the low-byte of the result

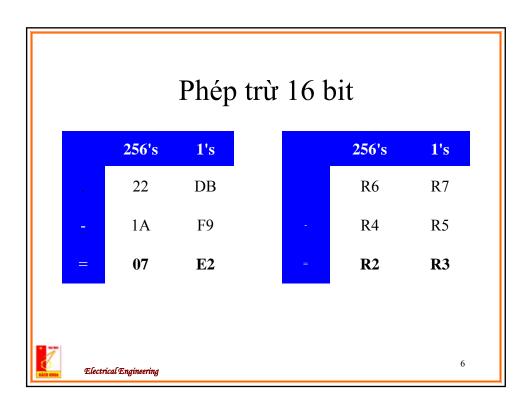
MOV A,R6 ;Move the high-byte into the accumulator ADDC A,R4 ;Add the second high-byte to the accumulator, plus MOV R2,A ;Move the answer to the high-byte of the result

MOV A,#00h ;By default, the highest byte will be zero. ADDC A,#00h ;Add zero, plus carry from step 2. MOV R1,A ;Move the answer to the highest byte of the



Electrical Engineering





Hàm trừ 16 bit

```
SUBB16_16:

;Step 1 of the process

MOV A,R7 ;Move the low-byte into the accumulator

CLR C ;Always clear carry before first subtraction

SUBB A,R5 ;Subtract the second low-byte from the accumulator

MOV R3,A ;Move the answer to the low-byte of the result

;Step 2 of the process

MOV A,R6 ;Move the high-byte into the accumulator

SUBB A,R4 ;Subtract the second high-byte from the accumulator

MOV R2,A ;Move the answer to the low-byte of the result

;Return - answer now resides in R2, and R3.

RET
```



Electrical Engineering

7

Cách dùng hàm trừ

```
;Load the first value into R6 and R7
MOV R6,#22h
MOV R7,#0DBh

;Load the second value into R4 and R5
MOV R4,#1Ah
MOV R5,#0F9h

;Call the 16-bit subtraction routine
LCALL SUBB16 16
```



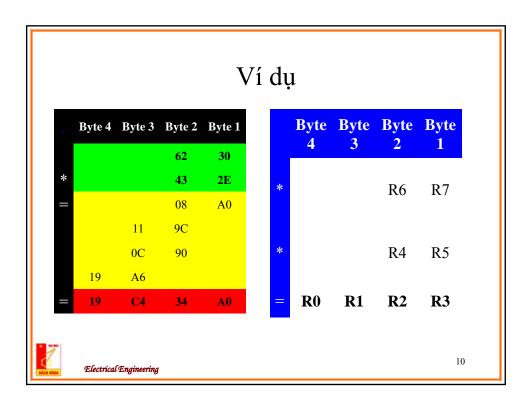
Electrical Engineering

Phép nhân 2 số 16 bit

- Nhân 2 số 16 bit tạo ra số 32 bit
- Ví dụ nhân 2 số:
 - **25,136** x **17,198**. = 432,288,928
 - $-6230h \times 432Eh = 19,C4,32,48h$



Electrical Engineering



Thuật toán

- Nhân R5 và R7, kết quả 16-bit lưu trong R2 vàR3.
- Nhân R5 và R6, cộng kết quả 16-bit vào R1 và R2.
- Nhân R4 và R7, cộng kết quả 16-bit vào R1 và R2.
- Nhân R4 và R6, cộng kết quả 16-bit vào R0 và R1



Electrical Engineering

1

Kết quả

```
MOV A,R5 ;Move the R5 into the Accumulator MOV B,R7 ;Move R7 into B
MUL AB ;Multiply the two values
MOV R2,B ;Move B (the high-byte) into R2
MOV R3,A ;Move A (the low-byte) into R3
```

```
MOV A,R5
            ; Move R5 back into the Accumulator
MOV B,R6
            ;Move R6 into B
MUL AB
            ;Multiply the two values
ADD A,R2
            ; Add the low-byte into the value already in R2
MOV R2,A
            ;Move the resulting value back into R2
MOV A,B
            ; Move the high-byte into the \operatorname{accumulator}
ADDC A, #00h ; Add zero (plus the carry, if any)
MOV R1,A
           ; Move the resulting answer into R1
MOV A, #00h ; Load the accumulator with zero
ADDC A, #00h ; Add zero (plus the carry, if any)
MOV RO,A
            ; Move the resulting answer to RO.
```

8

Electrical Engineering

Nhân 16 bít (tiếp)

```
MOV A, R4
           ;Move R4 into the Accumulator
MOV B,R7
           ;Move R7 into B
MUL AB
           ;Multiply the two values
ADD A,R2 ;Add the low-byte into the value already in R2
MOV R2,A
           ;Move the resulting value back into R2
MOV A, B
           ;Move the high-byte into the accumulator
ADDC A,R1 ;Add the current value of R1 (plus any carry)
MOV R1,A
           ;Move the resulting answer into R1.
MOV A, #00h ; Load the accumulator with zero
ADDC A,R0 ;Add the current value of R0 (plus any carry)
\mbox{MOV RO,A} \mbox{\ \ ;} \mbox{Move the resulting answer to R1.}
```

```
MOV A,R4 :Move R4 back into the Accumulator
MOV B,R6 :Move R6 into B
MUL AB :Multiply the two values
ADD A,R1 :Add the low-byte into the value already in R1
MOV R1,A :Move the resulting value back into R1
MOV A,B :Move the high-byte into the accumulator
ADDC A,R0 :Add it to the value already in R0 (plus any carry)
MOV R0,A :Move the resulting answer back to R0
```



Electrical Engineering

13

Cách sử dụng

```
;Load the first value into R6 and R7 MOV R6,#62h MOV R7,#30h ;Load the first value into R4 and R5 MOV R4,#43h MOV R5,#2Eh ;Call the 16-bit subtraction routine LCALL MUL16_16
```



Electrical Engineering

Phép chia 16 bit



Electrical Engineering

14

Dịch trái

```
MOV B,#00h ;Clear B since B will count the number of left-shifted bits div1:

INC B ;Increment counter for each left shift

MOV A,R2 ;Move the current divisor low byte into the accumulator RLC A ;Shift low-byte left, rotate through carry to apply highest bit to high-byte

MOV R2,A ;Save the updated divisor low-byte

MOV A,R3 ;Move the current divisor high byte into the accumulator RLC A ;Shift high-byte left high, rotating in carry from low-byte

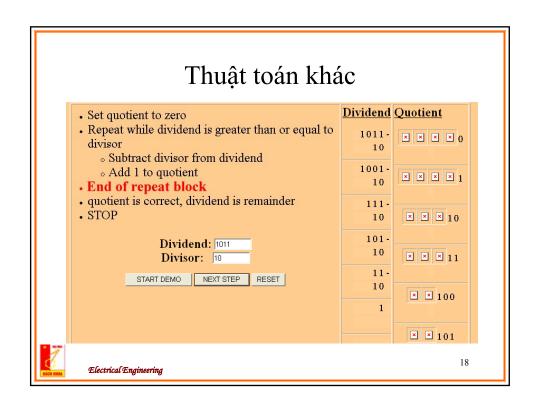
MOV R3,A ;Save the updated divisor high-byte

JNC div1 ;Repeat until carry flag is set from high-byte
```



Electrical Engineering

```
Phép trừ lặp lại
                      ;Shift right the divisor
div2:
   MOV A,R3
                      ;Move high-byte of divisor into accumulator
                     ;Rotate high-byte of divisor right and into carry ;Save updated value of high-byte of divisor ;Move low-byte of divisor into accumulator ;Rotate low-byte of divisor right, with carry from high-
   RRC A
   MOV R3, A
   MOV A,R2
RRC A
   MOV R2, A ;Save updated value of low-byte of divisor CLR C ;Clear carry, we don't need it anymore MOV 07h,R1 ;Make a safe copy of the dividend high-byte MOV 06h,R0 ;Make a safe copy of the dividend low-byte MOV A,R0 ;Move low-byte of dividend into accumulator
   SUBB A, R2
                     ;Dividend - shifted divisor = result bit (no factor, only
0 or 1)
MOV RO,A
MOV A,R1
                     ;Save updated dividend
;Move high-byte of dividend into accumulator
   SUBB A, R3 ; Subtract high-byte of divisor (all together 16-bit
substraction)
   MOV R1, A
JNC div3
                    ;Save updated high-byte back in high-byte of divisor
   JNC div3 ;If carry flag is NOT set, result is 1 MOV R1,07h ;Otherwise result is 0, save copy of divisor to undo
subtraction
MOV RO,06h
   CPL C
                     ;Invert carry, so it can be directly copied into result
   MOV A,R4
RLC A
MOV R4,A
MOV A,R5
                     ;Shift carry flag into temporary result
   RLC A
MOV R5, A
   DJNZ B, div2 ; Now count backwards and repeat until "B" is zero
Electrical Engineering
```



Bài tập

- Khởi tạo kết quả = 0
- while
 - Nếu số bị chia còn lớn hơn hoặc bằng thương số
 - Trừ số bị chia cho thương số
 - Tăng kết quả lên 1
- Kết thúc:
 - Số bị chia còn lại là phần dư



Electrical Engineering