RA2011026020065
Bharathwaj.M

STUDY EXPT

SRM INSTITUTE OF SCIENCE & TECHNOLOGY

## Expt: Study and program to carry save multiplication

* **Carry-save Addition:**

Consider adding six set of numbers (4 bits each in the example).

- The numbers are 1001, 0110, 1111, 0111, 1010, 0110 (all +ve).
- one way is to add them pair wise, getting three results, and then adding them again.

```
1001   1111   1010   0111   100101
0110   0111   0110   1010   10000
0111   1010   10000  100101 110101
```

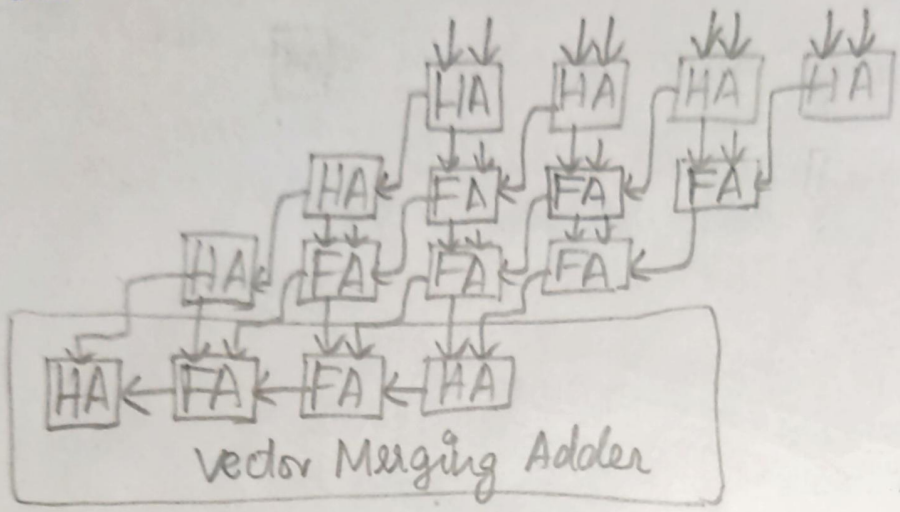
- other method is add them three at a time by saving carry.

```
1001   0111   00000  010101   001101
0110   1010   11110  010100   101000
1111   0110   01011  001100   110101
00000  01011  010101 001101   SUM
11110  01100  010100 101000   CARRY
```

* **Carry-save Multiplication:**

- n-bit carry-save adder take 2FA time for any n.
- for n x n bit multiplication, n or n/2 (for 2-bit at time of Booth's encoding) partial product can be generated.
- for n partial product, n/3 n-bit carry-save adders can be used.
- This yields 2n/3 partial results.
- Repeat this operation, until only two partial results are remaining / left.
- Add them using an appropriate size adder to obtain 2n bit result.
- For n=32, you need 80 carry-save adders in eight stages taking 8T where T is time for one-bit full adder.

- then, you need one carry-propagate (or) carry-look-ahead adder.




Result:

Thus the study of carry-look-ahead adder has been done and components has be realized.

PROGRAM

SRM INSTITUTE OF SCIENCE & TECHNOLOGY

Expt: Program involving Arithmetic Instruction on 16-bit data subtraction

* Aim:
To implement assembly language program for subtraction of two 16 bit numbers.

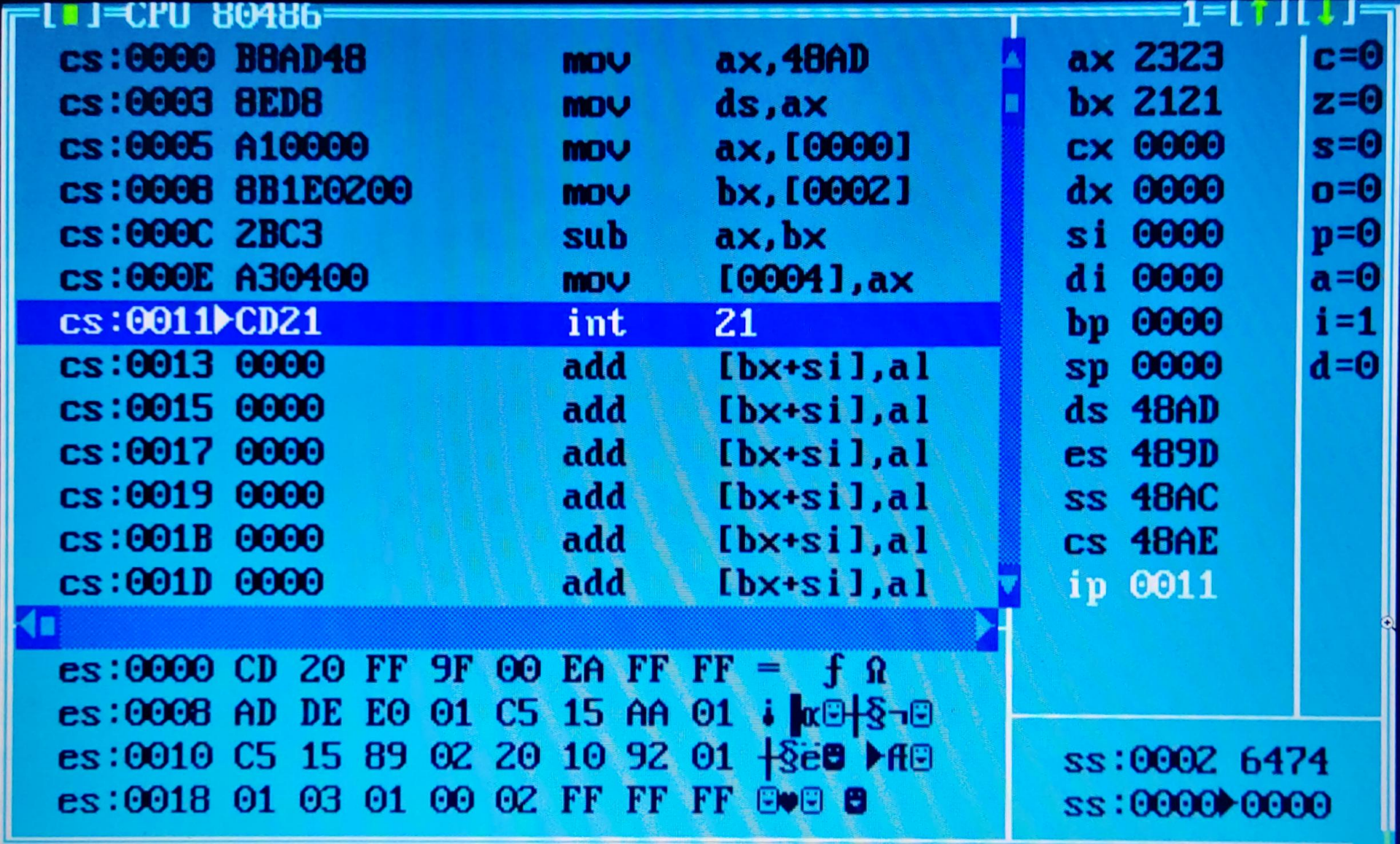
* Apparatus: TASM Software, P.C.

* Program:

```
DATA SEGMENT
N1 DW 4444H
N2 DW 2121H
RES DW ?
DATA ENDS

CODE SEGMENT
ASSUME CS: CODE, DS:DATA
START: MOV AX, DATA
MOV DS, AX
MOV AX, N1
MOV BX, N2
SUB AX, BX
MOV RES, AX
INT 21H
CODE ENDS
END START
```

* Result:
AX = 2323h


CPU 80486
cs:0000 B8AD48 mov ax,48AD
cs:0003 8ED8 mov ds,ax
cs:0005 A10000 mov ax,[0000]
cs:0008 8B1E0200 mov bx,[0002]
cs:000C 2BC3 sub ax,bx
cs:000E A30400 mov [0004],ax
cs:0011 CD21 int 21
cs:0013 0000 add [bx+si],al
cs:0015 0000 add [bx+si],al
cs:0017 0000 add [bx+si],al
cs:0019 0000 add [bx+si],al
cs:001B 0000 add [bx+si],al
cs:001D 0000 add [bx+si],al
ax 2323
bx 2121
cx 0000
dx 0000
si 0000
di 0000
bp 0000
sp 0000
ds 48AD
es 489D
ss 48AC
cs 48AE
ip 0011
c=0
z=0
s=0
o=0
p=0
a=0
i=1
d=0
es:0000 CD 20 FF 9F 00 EA FF FF
es:0008 AD DE E0 01 C5 15 AA 01
es:0010 C5 15 89 02 20 10 92 01
es:0018 01 03 01 00 02 FF FF FF
ss:0002 6474
ss:0000 0000