

Homework3 ARM与Thumb指令验证

姓名：胡成成

学号：

Problem

用ARM汇编设计程序，验证ARM处理器的ARM指令和Thumb指令下执行任意两个8位无符号数相乘结果的一致性。

Answer

本次实验采用keil环境测试，在ARM与THUMB指令的切换中需要使用CODE32与CODE16伪指令。

其中还跳转指令从ARM指令集切换到THUMB指令集。

```
AREA      ARMex, CODE, READONLY
ENTRY
CODE32
; MOV Opcode
MOV r1, #5
MOV r2, #2
; 32bit ARM
UMULL r3, r4, r1, r2
ADR r0, mul16 + 1
BX r0
CODE16
mul16
; MOV Opcode
MOV r4, #5
MOV r5, #2
; 16bit Thumb
MUL r5, r4
END
```

编译后在 `MOV r1 #5` 处下断点，此时通用寄存器：

Register	Value
Current	
R0	0x00000000
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000000
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x00000000
Mode	Supervisor
States	0
Sec	0.00000000
CP15	
CP15 - Cache	
CP15 - MMU	
CP15 - PID	

ProjectRegisters

```
15:      BX r0
16:
17:      CODE16
18: mull6
19:
```

operand_test.sHomework3-1.s

```
1      AREA      ARMex, CODE, READONLY
2      ENTRY
3
4      CODE32
5
6      ; MOV Opcode
7      MOV r1, #5
8      MOV r2, #2
9
10     ; 32bit ARM
11     UMULL r3, r4, r1, r2
12
13     ADR r0, mull6 + 1
14
15     BX r0
16
17     CODE16
18     mull6
19
20     ; MOV Opcode
21     MOV r4, #5
22     MOV r5, #2
23
24     ; 16bit Thumb
```

在ARM指令集下，将5和2分别放到r1，r2寄存器，如下图所示：

Register	Value
Current	
R0	0x00000000
R1	0x00000005
R2	0x00000002
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000008
CPSR	0x000000D3
SPSR	0x00000000
+ User/System	
+ Fast Interrupt	
+ Interrupt	
+ Supervisor	
+ Abort	
+ Undefined	
- Internal	
PC \$	0x00000008
Mode	Supervisor
States	2
Sec	0.00000017
+ CP15	
+ CP15 - Cache	
+ CP15 - MMU	
+ CP15 - PID	


```

11:      UMULL r3, r4, r1, r2
12:
⇒ 0x00000008 E0843291 UMULL R3,R
13:      ADR r0, mull6 + 1
14:
<

operand_test.s  Homework3-1.s

4      CODE32
5
6      ; MOV Opcode
7      MOV r1, #5
8      MOV r2, #2
9
10     ; 32bit ARM
11     UMULL r3, r4, r1, r2
12
13     ADR r0, mull6 + 1
14
15     BX r0
16
17     CODE16
18     mull6
19
20     ; MOV Opcode
21     MOV r4, #5
22     MOV r5, #2
23
24     ; 16bit Thumb
25     MUL r5, r4
26
27     END
28
<

```

乘法计算后，寄存器中值为10：

Register	Value
Current	
R0	0x00000000
R1	0x00000005
R2	0x00000002
R3	0x0000000A
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x0000000C
+ CPSR	0x000000D3
+ SPSR	0x00000000
+ User/System	
+ Fast Interrupt	
+ Interrupt	
+ Supervisor	
+ Abort	
+ Undefined	
- Internal	
PC \$	0x0000000C
Mode	Supervisor
States	5
Sec	0.00000042
+ CP15	
+ CP15 - Cache	
+ CP15 - MMU	
+ CP15 - PID	

13:	ADR r0, mull6 + 1
14:	
⇒ 0x0000000C	E28F0001 ADD R0, PC
15:	BX r0
16:	
<	
operand_test.s	Homework3-1.s
6	; MOV Opcode
7	MOV r1, #5
8	MOV r2, #2
9	
10	; 32bit ARM
11	UMULL r3, r4, r1, r2
12	
▶ 13	ADR r0, mull6 + 1
14	
15	BX r0
16	
17	CODE16
18	mull6
19	
20	; MOV Opcode
21	MOV r4, #5
22	MOV r5, #2
23	
24	; 16bit Thumb
25	MUL r5, r4
26	
27	END
28	
<	

切换到THUMB指令，再次将5和2装载到r4和r5中：

Registers

Register	Value
Current	
R0	0x00000015
R1	0x00000005
R2	0x00000002
R3	0x0000000A
R4	0x00000005
R5	0x00000002
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000018
CPSR	0x000000F3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x00000018
Mode	Supervisor
States	11
Sec	0.00000092
CP15	
CP15 - Cache	
CP15 - MMU	
CP15 - PID	

Project
Registers

Disassembly

```

25:          MUL r5, r4
0x00000018 4365      MUL      R5,
0x0000001A 0000      LSL      R0,
0x0000001C 0000      LSL      R0,
0x0000001E 0000      LSL      R0,

```

operand_test.s
Homework3-1.s

```

6          ; MOV Opcode
7          MOV r1, #5
8          MOV r2, #2
9
10         ; 32bit ARM
11         UMULL r3, r4, r1, r2
12
13         ADR r0, mull6 + 1
14
15         BX r0
16
17         CODE16
18         mull6
19
20         ; MOV Opcode
21         MOV r4, #5
22         MOV r5, #2
23
24         ; 16bit Thumb
25         MUL r5, r4
26
27         END
28

```

乘法计算后，寄存器r5中值为10：

Register	Value
Current	
R0	0x00000015
R1	0x00000005
R2	0x00000002
R3	0x0000000A
R4	0x00000005
R5	0x0000000A
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x0000001C
+ CPSR	0x000000F3
+ SPSR	0x00000000
+ User/System	
+ Fast Interrupt	
+ Interrupt	
+ Supervisor	
+ Abort	
+ Undefined	
- Internal	
PC \$	0x0000001C
Mode	Supervisor
States	14
Sec	0.00000117
+ CP15	
+ CP15 - Cache	
+ CP15 - MMU	
+ CP15 - PID	

25:	MUL r5, r4		
0x00000018	4365	MUL	R5,R
0x0000001A	0000	LSL	R0,R
→ 0x0000001C	0000	LSL	R0,R
0x0000001E	0000	LSL	R0,R
<			

operand_test.s

Homework3-1.s

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

; MOV Opcode

MOV r1, #5

MOV r2, #2

; 32bit ARM

UMULL r3, r4, r1, r2

ADR r0, mull6 + 1

BX r0

CODE16

mull6

; MOV Opcode

MOV r4, #5

MOV r5, #2

; 16bit Thumb

MUL r5, r4

END

<

验证发现，在两个指令集下无符号惩罚的计算结果一致。