

MPCA Assignment : Unit-2

- Name : P K Navin Shrinivas
- Section : D
- SRN : PES2UG20CS237

Question 1: Binary Search in ARM7RDMI Assembly

Code:

```
.data
    ARR: .WORD 2,4,7,9,10,14,15
    KEY: .WORD 14
    SIZE: .WORD 7
    OUTPUT1: .ASCIZ "Successful Search"
    OUTPUT2: .ASCIZ "Unsuccessful Search"
.text

LDR r11,=SIZE
LDR r0,[r11]
LDR r1,=ARR
ADD r2,r1,r0,LSL #2
LDR r11,=KEY
LDR r3,[r11]

initloop:
    SUB r6,r2,r1
    CMP r6,#4
    BEQ failed
```

```
SUB r4,r2,r1
MOV r4,r4,LSR #2
MOV r4,r4,LSR #1
MOV r4,r4,LSL #2
ADD r4,r4,r1
LDR r5,[r4]
CMP r5,r3
BEQ found
BLT gogreater
B golesser
```

gogreater:

```
MOV r1,r4
B initloop
```

golesser:

```
MOV r2,r4
B initloop
```

found:

```
ldr r1,=OUTPUT1
B LOOP
```

failed:

```
ldr r1,=OUTPUT2
B LOOP
```

LOOP:

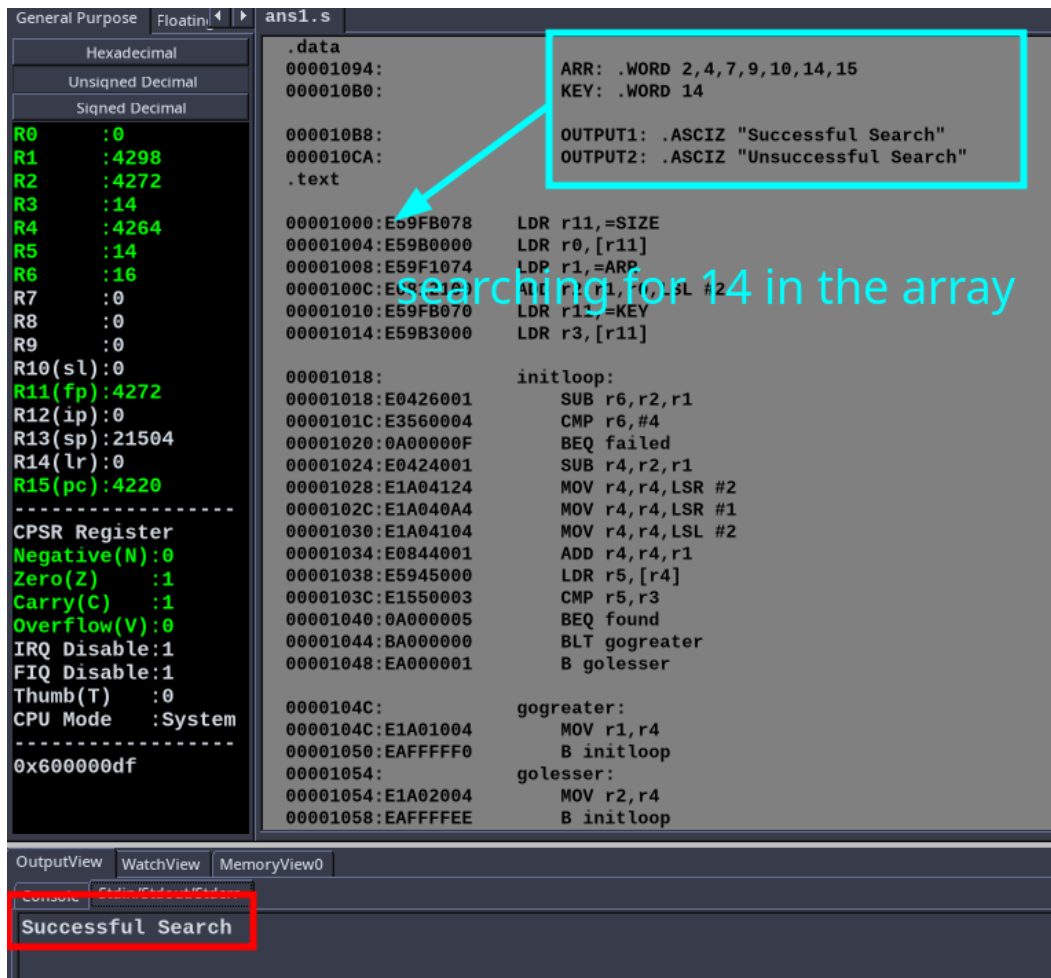
```
LDRB R0,[R1],#1
CMP R0,#0
SWINE 0x00
BNE LOOP
SWI 0x11
```

end: .end

Screenshots:

The screenshot displays a debugger interface with the following components:

- Registers Panel (Left):** Shows the state of various registers. R15 (PC) is 4220. The CPSR register shows Zero (Z) and Carry (C) flags are set to 1.
- Assembly View (Center):** Displays the assembly code for the program. The code includes an initialization loop and a search loop. The search loop is currently executing, with the instruction `ADD r7, r1, r0, LSL #2` highlighted. A red arrow points from the data section to this instruction, with the text "searching for 3 in the array" overlaid.
- Data Section (Top Right):** A red box highlights the data section where the array `ARR` is defined as `.WORD 2,4,7,9,10,14,15`. The `KEY` is defined as `.WORD 3`. The `OUTPUT1` and `OUTPUT2` strings are also defined.
- Console View (Bottom):** A red box highlights the console output, which shows the text `Unsuccessful Search`.



Question 2 : Search for a substring in a string using ARM7TDMI

Code:

```

.data
    STRING: .ASCIZ "My name is Bond"
    SUBSTR: .ASCIZ "name"
    OUTPUT1: .ASCIZ "String Present"
    OUTPUT2: .ASCIZ "String Absent"

.text

```

```
LDR r0,=STRING
LDR r1,=SUBSTR
```

```
initmatch:
```

```
    LDRB r2,[r0],#1
    LDRB r3,[r1]
    CMP r2,#0
    BEQ notfound
    CMP r2,r3
    MOV r4,r0
    ADD r5,r1,#1
    BEQ submatch
    B initmatch
```

```
submatch:
```

```
    LDRB r2,[r4],#1
    LDRB r3,[r5],#1
    CMP r3,#0
    BEQ found
    CMP r2,r3
    BEQ submatch
    B initmatch
```

```
found:
```

```
    LDR r1,=OUTPUT1
    B LOOP
```

```
notfound:
```

```
    LDR r1,=OUTPUT2
    B LOOP
```

```
LOOP:
```

```

LDRB R0, [R1], #1
CMP R0, #0
SWINE 0x00
BNE LOOP
SWI 0x11

```

Screenshots:

The screenshot shows a debugger interface with the following components:

- General Purpose Registers:** R0: 0, R1: 4256, R2: 32, R3: 0, R4: 4228, R5: 4241, R6: 0, R7: 0, R8: 0, R9: 0, R10(sl): 0, R11(fp): 0, R12(ip): 0, R13(sp): 21504, R14(lr): 0, R15(pc): 4200.
- CPSR Register:** Negative(N): 0, Zero(Z): 1, Carry(C): 1, Overflow(V): 0, IRQ Disable: 1, FIQ Disable: 1, Thumb(T): 0, CPU Mode: System.
- Assembly Code (ans2.s):**
 - .data:**

```

0000107C: STRING: .ASCIZ "My name is Bond"
0000108C: SUBSTR: .ASCIZ "name"
00001091: OUTPUT1: .ASCIZ "String Present"
000010A0: OUTPUT2: .ASCIZ "String Absent"

```
 - .text:**

```

00001000:E59F0064 LDR r0,=STRING
00001004:E59F1064 LDR r2,[r1],#1
00001008: initmatch:
00001008:E4D02001 LDRB r2,[r2],#1
0000100C:E5D13000 LDRB r3,[r1]
00001010:E3520000 CMP r2,#0
00001014:0A00000D BEQ notfound
00001018:E1520003 CMP r2,r3
0000101C:E1A04000 MOV r4,r0
00001020:E2815001 ADD r5,r1,#1
00001024:0A000000 BEQ submatch
00001028:EAF0FF66 B initmatch
0000102C: submatch:
0000102C:E4D42001 LDRB r2,[r4],#1
00001030:E4D53001 LDRB r3,[r5],#1
00001034:E3530000 CMP r3,#0
00001038:0A000002 BEQ found
0000103C:E1520003 CMP r2,r3
00001040:0AFF0FF9 BEQ submatch
00001044:EAF0FF6F B initmatch
00001048: found:
00001048:E59F1024 LDR r1,=OUTPUT1
0000104C:EA000001 B LOOP
00001050: notfound:
00001050:E59F1020 LDR r1,=OUTPUT2

```
- Console:** A red box highlights the output "String Present".

General Purpose Floating ans2.s

Hexadecimal
Unsigned Decimal
Signed Decimal

R0 :0
R1 :4271
R2 :0
R3 :74
R4 :4235
R5 :4237
R6 :0
R7 :0
R8 :0
R9 :0
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):21504
R14(lr):0
R15(pc):4200

CPSR Register
Negative(N):0
Zero(Z) :1
Carry(C) :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T) :0
CPU Mode :System

0x600000df

.data
0000107C: STRING: .ASCIZ "My name is Bond"
0000108C: SUBSTR: .ASCIZ "James"
00001092: OUTPUT1: .ASCIZ "String Present"
000010A1: OUTPUT2: .ASCIZ "String Absent"

.text
00001000:E59F0064 LDR r0,=STRING
00001004:E59F1064 LDR r1,=SUBSTR
00001008: initmatch:
00001008:E4D02001 LDRB r2,[r0],#1
0000100C:E5D13000 LDRB r3,[r1]
00001010:E3520000 CMP r2,#0
00001014:0A00000D BEQ notfound
00001018:E1520003 CMP r2,r3
0000101C:E1A04000 MOV r4,r0
00001020:E2815001 ADD r5,r1,#1
00001024:0A000000 BEQ submatch
00001028:EAF0FF66 B initmatch
0000102C: submatch:
0000102C:E4D42001 LDRB r2,[r4],#1
00001030:E4D53001 LDRB r3,[r5],#1
00001034:E3530000 CMP r3,#0
00001038:0A000002 BEQ found
0000103C:E1520003 CMP r2,r3
00001040:0AF0FF69 BEQ submatch
00001044:EAF0FF6F B initmatch
00001048: found:
00001048:E59F1024 LDR r1,=OUTPUT1
0000104C:EA000001 B LOOP
00001050: notfound:
00001050:E59F1020 LDR r1,=OUTPUT2

searching for James in the string

OutputView WatchView MemoryView0
Console
String Absent

Question-3

- ① LDR R1, [R2, #40] F D E M W
 ② ADD R2, R3, R3 F D E M W
 ③ ADD R1, R2, R2 F D E M W
 ④ STR R1, [R2, #20] F D E M W

~~without forwarding:~~

- a) ~~① & ② : Anti dependence on R2
 ② & ③ : true dependence on R2
 ② & ③ : true dependence on R1
 ③ & ④ : output dependence on R1
 ① & ④ : " " "
 ① & ③ : Output " " "~~

a) without forwarding.

- ① & ② : Anti dependence on R2
 ① & ③ : Output dependence on R1
 ① & ④ : true dependence on R1
 if no partitioning.
 ② & ③ : true dependence on R2
 ② & ④ : true dependence on R2
 ③ & ④ : true dependence on R1

b) Considering Separate I-cache & D-cache
 with fwding. without:

① & ③

① ③

① ④ if no partition

② ③

② ④

③ ④

- c)
- ① ins
 - ② ins
 - Nop (mem access)
 - ③ ins
 - ④ ins.

→ Question: 4

a)

LDR	<u>F</u>	<u>D</u>	<u>E</u>	<u>M</u>	<u>W</u>
BEQ	<u>F</u>	<u>D</u>	<u>E</u>	<u>M</u>	<u>W</u>
ADD	discarded → <u>F</u> <u>D</u> <u>Not executed</u>				
BEQ	<u>F</u>	<u>D</u>	<u>E</u>	<u>M</u>	<u>W</u>
STR	<u>F</u>	<u>D</u>	<u>E</u>	<u>M</u>	<u>W</u>
AND	<u>F</u>	<u>D</u>	<u>E</u>	<u>M</u>	<u>W</u>

b) with delay: (Branch logic in EX stage)

LDR	<u>F</u>	<u>D</u>	<u>E</u>	<u>M</u>	<u>W</u>
BEQ	<u>F</u>	<u>D</u>	<u>E</u>	<u>M</u>	<u>W</u>

SAFE: INSTRUCTION.

SAFE INSTRUCTION.

BEQ	<u>F</u>	<u>D</u>	<u>E</u>	<u>M</u>	<u>W</u>
STR	<u>F</u>	<u>D</u>	<u>E</u>	<u>M</u>	<u>W</u>
	<u>F</u>	<u>D</u>	<u>E</u>	<u>M</u>	<u>W</u>