

ARM JOURNAL

ARM7TDMI

PAVAN C E
01FE17BEC114
DIV:B(BATCH B2)
ROLL NO:139

//ASM TO FIND SUM OF EVEN NUMBERS IN A GIVEN ARRAY OF 5 ELEMENTS

CODE 1:

```

AREA SUMEVEN, CODE
ENTRY
MOV R3, #0
MOV R5, #05
LDR R0, =0X40000000
EVEN LDR R2, [R0], #4
MOVS R6, R2, ROR #01
ADDCC R3, R3, R2
SUBS R5, R5, #1
BNE EVEN
STOP B STOP
END

```

The screenshot displays the uVision4 IDE interface with the following components:

- Registers:** A table showing the state of registers R0 through R15. R0 is 0x40000014, R1 is 0x00000000, R2 is 0x00000007, R3 is 0x0000000a, R4 is 0x00000000, R5 is 0x00000000, R6 is 0x80000003, R7 is 0x00000000, R8 is 0x00000000, R9 is 0x00000000, R10 is 0x00000000, R11 is 0x00000000, R12 is 0x00000000, R13 (SP) is 0x00000000, R14 (LR) is 0x00000000, R15 (PC) is 0x00000020. CPSR is 0x600000d3. Flags N=0, Z=1, C=1.
- Disassembly:** Shows the assembly code being executed. The current instruction is `11: STOP B STOP` at address `0x00000020`. Other instructions include `8: ADDCC R3, R3, R2`, `9: SUBS R5, R5, #1`, and `10: BNE EVEN`.
- Command:** Displays the code size limit and current usage. It states: "Running with Code Size Limit: 32K", "Load 'C:\Users\exam\Desktop\asm\project1.AXF'", "*** Restricted Version with 32768 Byte Code Size Limit", and "*** Currently used: 36 Bytes (0%)".
- Memory:** Shows a memory dump starting at address `0x40000000`. The dump shows a sequence of bytes: `02 00 00 00 05 00 00 00 06 00 00 00 02 00 00 00 07 00 00 00 00 00 00`.
- Status Bar:** Indicates "Real-Time Agent: Target Stopped", "Simulation", and "t1: 0.00000383 sec". The date and time are "10:45 AM 2/16/2019".

CODE 2:

AREA SUMEVEN2, CODE

ENTRY

MOV R3, #0

MOV R5, #05

MOV R1, #00000001

LDR R0, =0X40000000

EVEN LDR R2, [R0], #4

MOV R6, R2

AND R8, R1, R6

MOVS R8, R8, LSR #1

ADDCC R3, R3, R2

SUBS R5, R5, #1

BNE EVEN

STOP B STOP

END

The screenshot displays the Keil uVision4 IDE with the following components:

- Register Window:** Shows the current state of registers R0 through R15 and CPSR. R0 is 0x40000014, R1 is 0x00000001, R2 is 0x00000003, R3 is 0x00000004, R4 is 0x00000000, R5 is 0x00000000, R6 is 0x00000003, R7 is 0x00000000, R8 is 0x00000000, R9 is 0x00000000, R10 is 0x00000000, R11 is 0x00000000, R12 is 0x00000000, R13 (SP) is 0x00000000, R14 (LR) is 0x00000000, R15 (PC) is 0x0000002c, and CPSR is 0x600000d3.
- Disassembly Window:** Shows the assembly code being executed, including instructions like SUBS R5, R5, #1, BNE EVEN, STOP B STOP, and LDR R2, [R0], #4.
- Source Window (EVEN_SUM2.ASM):** Shows the original assembly code being assembled, matching the code provided in the question.
- Command Window:** Displays the command line output, including the code size limit (32K) and the current usage (48 Bytes).
- Memory Window:** Shows the memory dump starting at address 0x40000000, displaying hexadecimal and binary data.

//ASM TO FIND SUM OF ODD NUMBERS IN A GIVEN ARRAY OF 5 ELEMENTS

CODE 1:

```
AREA SUMODD, CODE
ENTRY
MOV R3, #0
MOV R5, #05
LDR R0, =0X40000000
ODD LDR R2, [R0], #4
MOVS R6, R2, ROR #01
ADDCS R3, R3, R2
SUBS R5, R5, #1
BNE ODD
STOP B STOP
END
```

The screenshot shows the uVision4 IDE interface. The main window displays the assembly code for 'ODD_SUM.ASM'. The registers window on the left shows the state of R0 through R15 and CPSR. The command window at the bottom shows the execution progress and memory dump.

Registers:

Register	Value
R0	0x40000014
R1	0x00000000
R2	0x00000001
R3	0x00000009
R4	0x00000000
R5	0x00000000
R6	0x80000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000020
CPSR	0x600000d3
N	0
Z	1
C	1

Disassembly:

```

11: STOP B STOP
0x00000020 EAffffff B 0x00000020
0x00000024 00000000 ANDEQ R0, R0, R0
0x00000028 00000000 ANDEQ R0, R0, R0
0x0000002C 00000000 ANDEQ R0, R0, R0
0x00000030 00000000 ANDEQ R0, R0, R0
0x00000034 00000000 ANDEQ R0, R0, R0
0x00000038 00000000 ANDEQ R0, R0, R0
0x0000003C 00000000 ANDEQ R0, R0, R0
0x00000040 00000000 ANDEQ R0, R0, R0
0x00000044 00000000 ANDEQ R0, R0, R0

```

Assembly Code:

```

04 0.083 µs MOV R5, #05
05 0.083 µs LDR R0, =0X40000000
06 1.250 µs ODD LDR R2, [R0], #4
07 0.417 µs MOVS R6, R2, ROR #01
08 0.417 µs ADDCS R3, R3, R2
09 0.417 µs SUBS R5, R5, #1
10 1.083 µs BNE ODD
11 STOP B STOP
END

```

Command:

```

Running with Code Size Limit: 32K
Load "C:\Users\exam\Desktop\arm\project1.AXF"

*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 36 Bytes (0%)

```

Memory 1:

Address: 0X40000000

```

0x40000000: 03 00 00 00 05 00 00 00 02 00 00 00 04 00 00 00 01 00 00 00 00 00 00
0x40000017: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000002E: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000045: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000005C: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000073: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000008A: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x400000A1: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

Bottom Bar:

Real-Time Agent: Target Stopped | Simulation | t1: 0.0000383 sec | CAP NUM SCRL OVR R/W | 10:48 AM 2/16/2019

CODE 2:

AREA SUMODD2, CODE

ENTRY

MOV R3, #0

MOV R5, #05

MOV R1, #00000001

LDR R0, =0X40000000

ODD LDR R2, [R0], #4

MOV R6, R2

AND R8, R1, R6

MOVS R8, R8, LSR #1

ADDCS R3, R3, R2

SUBS R5, R5, #1

BNE ODD

STOP B STOP

END

The screenshot displays the Keil uVision4 IDE with the following components:

- Registers:** A list of registers (R0-R15, CPSR, SPSR) and their current values. R0 is 0x40000014, R1 is 0x00000001, R2 is 0x00000005, R3 is 0x00000009, R4 is 0x00000000, R5 is 0x00000000, R6 is 0x00000005, R7 is 0x00000000, R8 is 0x00000000, R9 is 0x00000000, R10 is 0x00000000, R11 is 0x00000000, R12 is 0x00000000, R13 (SP) is 0x00000000, R14 (LR) is 0x00000000, R15 (PC) is 0x0000002c, CPSR is 0x600000d3, and SPSR is 0x00000000.
- Disassembly:** Shows the assembly code with addresses and timing. The current instruction is at address 0x40000000: `0.417 μs 0x00000024 E2555001 SUBS R5, R5, #1`. Other instructions include `BNE ODD`, `STOP B STOP`, and `EAFFFFFF B`.
- Source:** Shows the assembly code with comments. The current instruction is at address 0x40000000: `0.417 μs 0x00000024 E2555001 SUBS R5, R5, #1`. Other instructions include `BNE ODD`, `STOP B STOP`, and `EAFFFFFF B`.
- Command:** Shows the command line: `Running with Code Size Limit: 32K`. It also displays the load path: `Load "C:\Keil\ARM\Examples\Measure\keel03.AMF"`. Below this, it states: `*** Restricted Version with 32768 Byte Code Size Limit` and `*** Currently used: 48 Bytes (0%)`.
- Memory:** Shows the memory address 0x40000000 and the corresponding memory contents. The memory is filled with zeros.

The bottom status bar indicates: `Real-Time Agent: Target Stopped`, `Simulation`, `t1: 0.00000550 sec`, `L:14 C:1`, `CAP NUM SCRL OVR R/W`, `0003`, `17-02-2019`.

```
//ASM TO FIND THE LARGEST NUMBER IN THE GIVEN ARRAY
```

```
CODE 1:AREA LARGE,CODE
```

```
ENTRY
```

```
LDR R0,=0X40000000
```

```
MOV R5,#5
```

```
LDR R2,[R0],#4
```

```
LRG LDR R4,[R0],#4
```

```
CMP R2,R4
```

```
MOVCC R2,R4
```

```
SUBS R5,R5,#1
```

```
BNE LRG
```

```
STOP B STOP
```

```
END
```

The screenshot displays the uVision4 IDE with the following components:

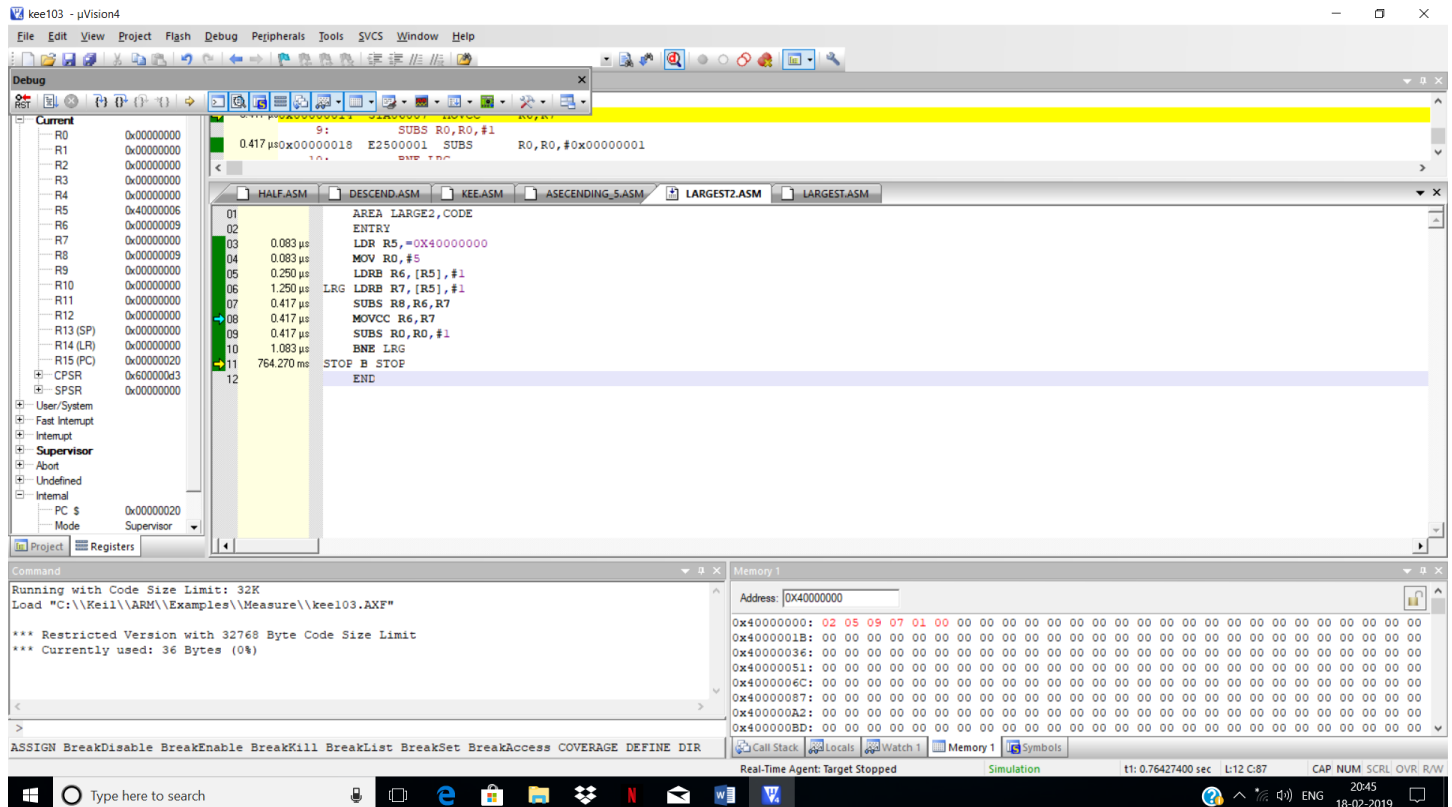
- Registers:** A list of registers (R0-R15, CPSR) with their current values. R5 is highlighted, showing a value of 0x00000000.
- Disassembly:** Shows the assembly code being executed. The current instruction is `1.000 μs 0x0000001C 1AFFFFFFA BNE 0x0000000C`, which corresponds to the `BNE LRG` instruction in the source code. The instruction list includes:
 - 6: LRG LDR R4,[R0],#4
 - 7: CMP R2,R4
 - 8: MOVCC R2,R4
 - 9: SUBS R5,R5,#1
 - 10: BNE LRG
 - 11: STOP B STOP
- Command:** Shows the command `Running with Code Size Limit: 32K` and the load path `Load "C:\Users\exam\Desktop\arm\project1.AXF"`. It also displays the code size limit and usage: `*** Restricted Version with 32768 Byte Code Size Limit` and `*** Currently used: 36 Bytes (0%)`.
- Memory:** Shows the memory address `0x40000000` and its contents, which are all zeros.
- Status Bar:** Indicates `Real-Time Agent: Target Stopped` and `Simulation` mode. The time is `tt: 0.00000392 sec` and the clock is `L:10 C:1`. The date and time are `11:35 AM 2/16/2019`.

CODE 2:

```

AREA LARGE2, CODE
ENTRY
LDR R5, =0x40000000
MOV R0, #5
LDRB R6, [R5], #1
LRG LDRB R7, [R5], #1
SUBS R8, R6, R7
MOVCC R6, R7
SUBS R0, R0, #1
BNE LRG
STOP B STOP
END

```



//ASM TO FIND THE SMALLEST NUMBER IN THE GIVEN ARRAY

```

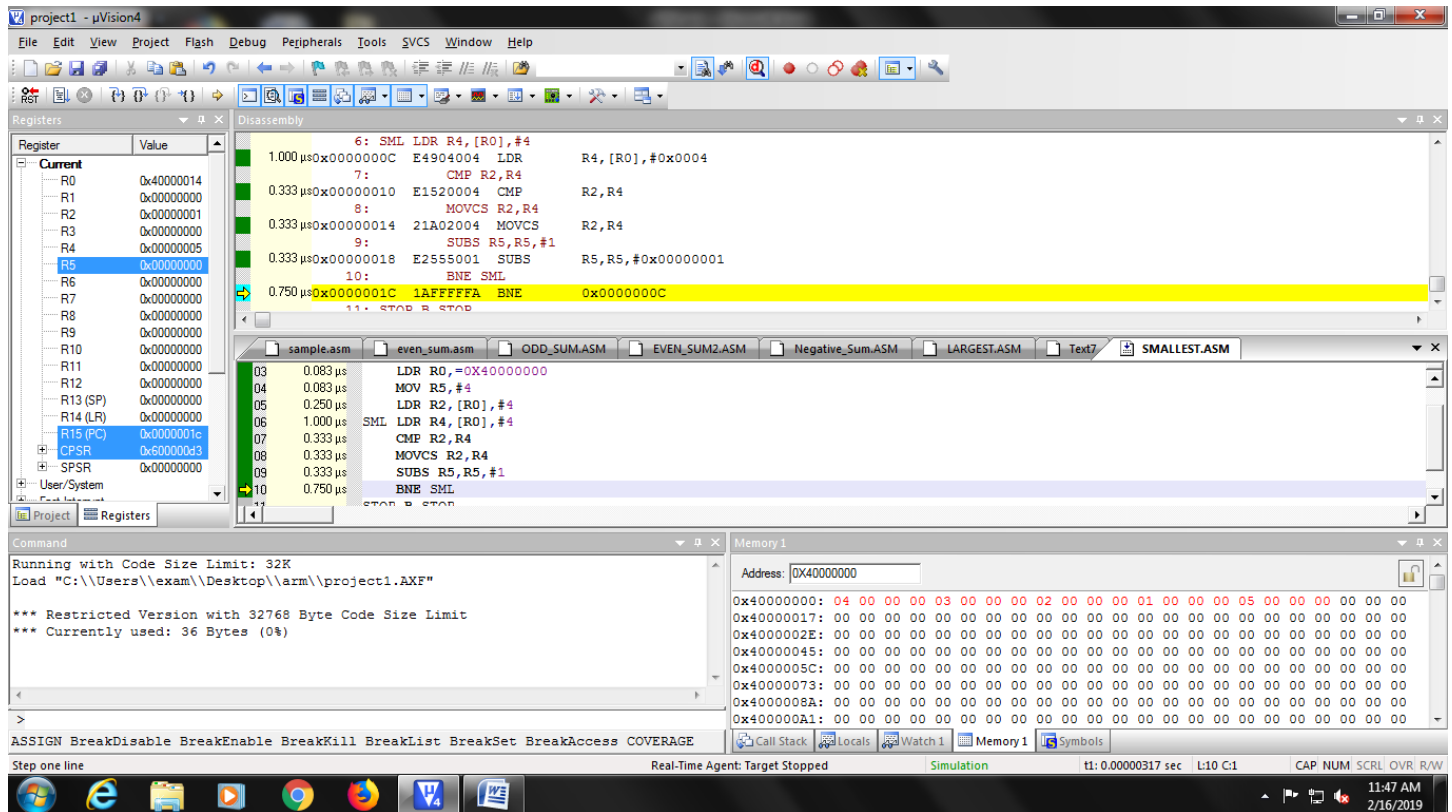
CODE 1: AREA SMALL, CODE
ENTRY

```

```

LDR R0,=0X40000000
MOV R5,#4
LDR R2,[R0],#4
SML LDR R4,[R0],#4
CMP R2,R4
MOVCS R2,R4
SUBS R5,R5,#1
BNE SML
STOP B STOP
END

```



CODE 2:

```

AREA SMALL2, CODE
ENTRY
LDR R5,=0X40000000
MOV R0,#5
LDRB R6,[R5],#1
SML LDRB R7,[R5],#1
SUBS R8,R6,R7
MOVCS R6,R7
SUBS R0,R0,#1
BNE SML
STOP B STOP
END

```

keel03 - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Debug

Current

- R0 0x00000000
- R1 0x00000000
- R2 0x00000000
- R3 0x00000000
- R4 0x00000000
- R5 0x40000000
- R6 0x00000000
- R7 0x00000000
- R8 0x00000012
- R9 0x00000000
- R10 0x00000000
- R11 0x00000000
- R12 0x00000000
- R13 (SP) 0x00000000
- R14 (LR) 0x00000000
- R15 (PC) 0x00000020
- CPSR 0x60000043
- SPSR 0x00000000
- User/System
- Fast Interrupt
- Interrupt
- Supervisor
- Abort
- Undefined
- Internal
- PC \$ 0x00000020
- Mode Supervisor

Project Registers

Command

Running with Code Size Limit: 32K
Load "C:\\Keil\\ARM\\Examples\\Measure\\keel03.AXF"

*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 36 Bytes (0%)

ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR

Memory 1

Address: 0x40000000

0x40000000: 56 45 12 58 99 00
0x4000001B: 00
0x40000036: 00
0x40000051: 00
0x4000006C: 00
0x40000087: 00
0x400000A2: 00
0x400000BD: 00

Call Stack Locals Watch 1 Memory 1 Symbols

Real-Time Agent: Target Stopped Simulation t1: 1.52598975 sec L:11 C:1 CAP NUM SCRL OVR R/W

2050
18-02-2019

//FIBONACCI SERIES OF 10 NUMBERS

```

AREA FIBONACCI, CODE
ENTRY
MOV R10, #10
LDR R3, =0x40000000
MOV R0, #0
SUB R1, R0, #1
MOV R2, #1
ADD R6, R2, R1
STR R6, [R3], #4
FIB ADD R6, R2, R0
    STR R6, [R3], #4
    MOV R2, R0
    MOV R0, R6
    SUBS R10, R10, #1
    BNE FIB
STOP B STOP
END

```

The screenshot displays the uVision4 IDE interface with the following components:

- Registers:** A list of registers (R0-R15, CPSR, SPSR) with their current values. R15 (PC) is highlighted at 0x00000034.
- Disassembly:** Shows assembly instructions from address 15 to 17. Instruction 15 is `BNE FIB` at 2.333 μs. Instruction 16 is `STOP B STOP` at 2.333 μs. Instruction 17 is `END` at 2.333 μs. A yellow highlight is on instruction 15.
- Command:** Shows the code size limit (32K) and the current usage (56 Bytes (0%)).
- Memory:** Shows the address 0x40000000 and a memory dump of 0x40000000 to 0x400000A1.
- Status Bar:** Shows "Real-Time Agent: Target Stopped", "Simulation", "t1: 0.00000800 sec", "L16 C1", "CAP NUM SCRL OVR R/W", and the time "12:07 PM 2/16/2019".

```
//TO ADD 2 ARRAY ELEMENTS AND STORE IN THIRD ARRAY
```

```
AREA SUMOFTWO_INTHIRD.CODE
```

```
ENTRY
```

```
LDR R3,=0x40000000
```

```
LDR R2,=AR1
```

```
LDR R1,=AR2
```

```
MOV R5,#5
```

```
AR3 LDR R4,[R2],#4
```

```
LDR R6,[R1],#4
```

```
ADD R7,R4,R6
```

```
STR R7,[R3],#4
```

```
SUBS R5,R5,#1
```

```
BNE AR3
```

```
STOP B STOP
```

```
AR1 DCD 0x01,0x02,0x04,0x06,0x05
```

```
AR2 DCD 0x01,0x02,0x04,0x06,0x05
```

```
END
```

The screenshot displays the uVision4 IDE interface with the following components:

- Registers:** A table showing the current values of registers R0 through R15, CPSR, and SPSR. R3 is highlighted with a value of 0x40000000.
- Disassembly:** A list of assembly instructions with their corresponding addresses and timing. The instruction at address 0x2555001, `SUBS R5,R5,#1`, is highlighted in yellow.
- Command:** A text area showing the execution status: "Running with Code Size Limit: 32K", "Load 'C:\Users\exam\Desktop\arm\project1.AXF'", and "Restricted Version with 32768 Byte Code Size Limit".
- Memory:** A memory dump window showing the contents of memory starting at address 0x40000000. The dump shows a sequence of zeros.
- Status Bar:** At the bottom, it indicates "Real-Time Agent: Target Stopped", "Simulation", and the time "12:17 PM 2/16/2019".

//DIVISION USING REPEATED SUBTRACTION

AREA A0, CODE

ENTRY

MOV R5, #15

MOV R6, #3

CMP R6, #0

BEQ EN

CMP R6, R5

MOVPL R6, #1

BPL EN

DIV SUB R7, R5, R6

ADD R1, R1, #1

CMP R7, R6

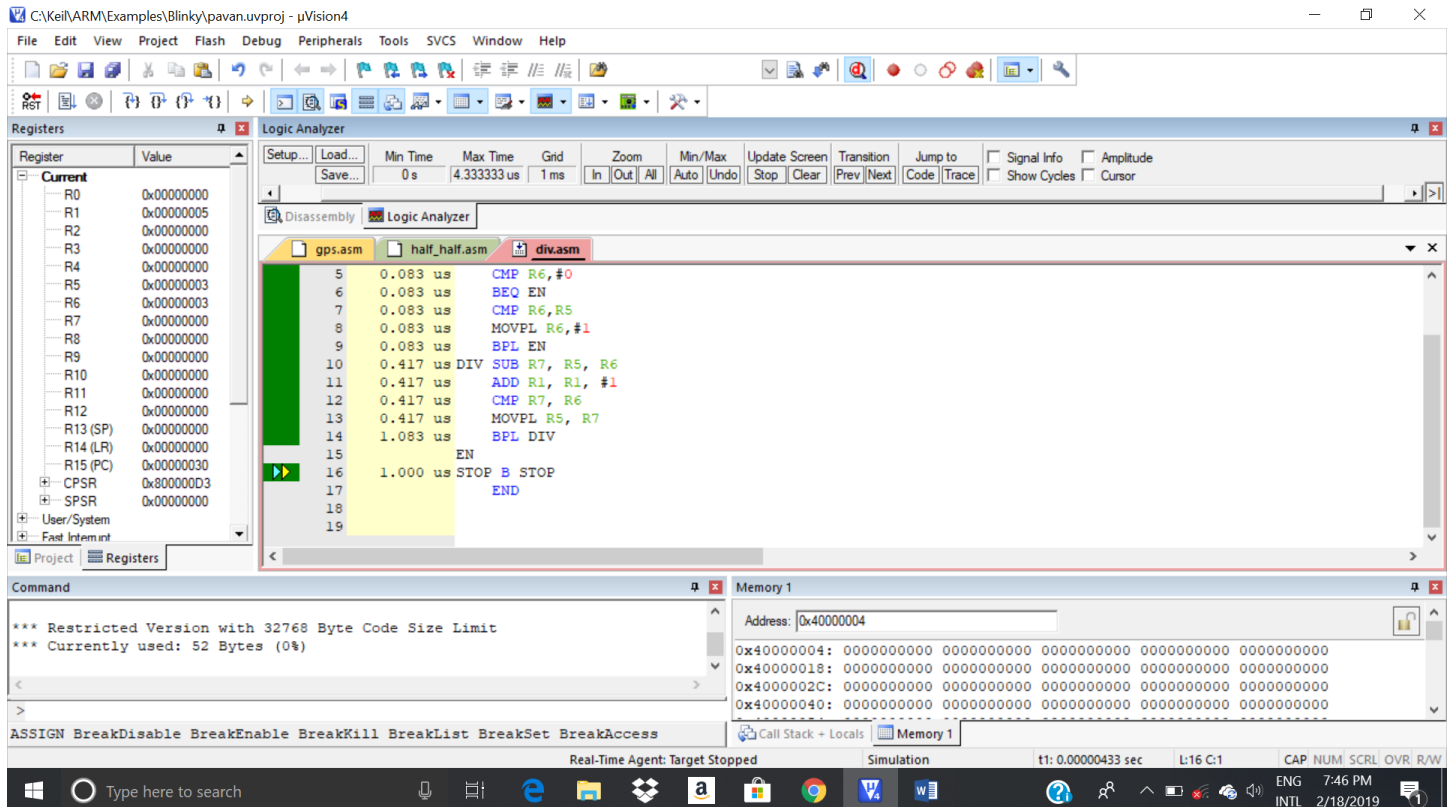
MOVPL R5, R7

BPL DIV

EN

STOP B STOP

END



CODE 2: DIVISION USING SHIFTING METHOD

AREA A0, CODE

ENTRY

MOV R1, #15

MOV R2, #3

CMP R2, #0

BEQ DEN

MOV R0, #0

MOV R3, #1

START CMP R2, R1

MOVL R2, R2, LSL #1

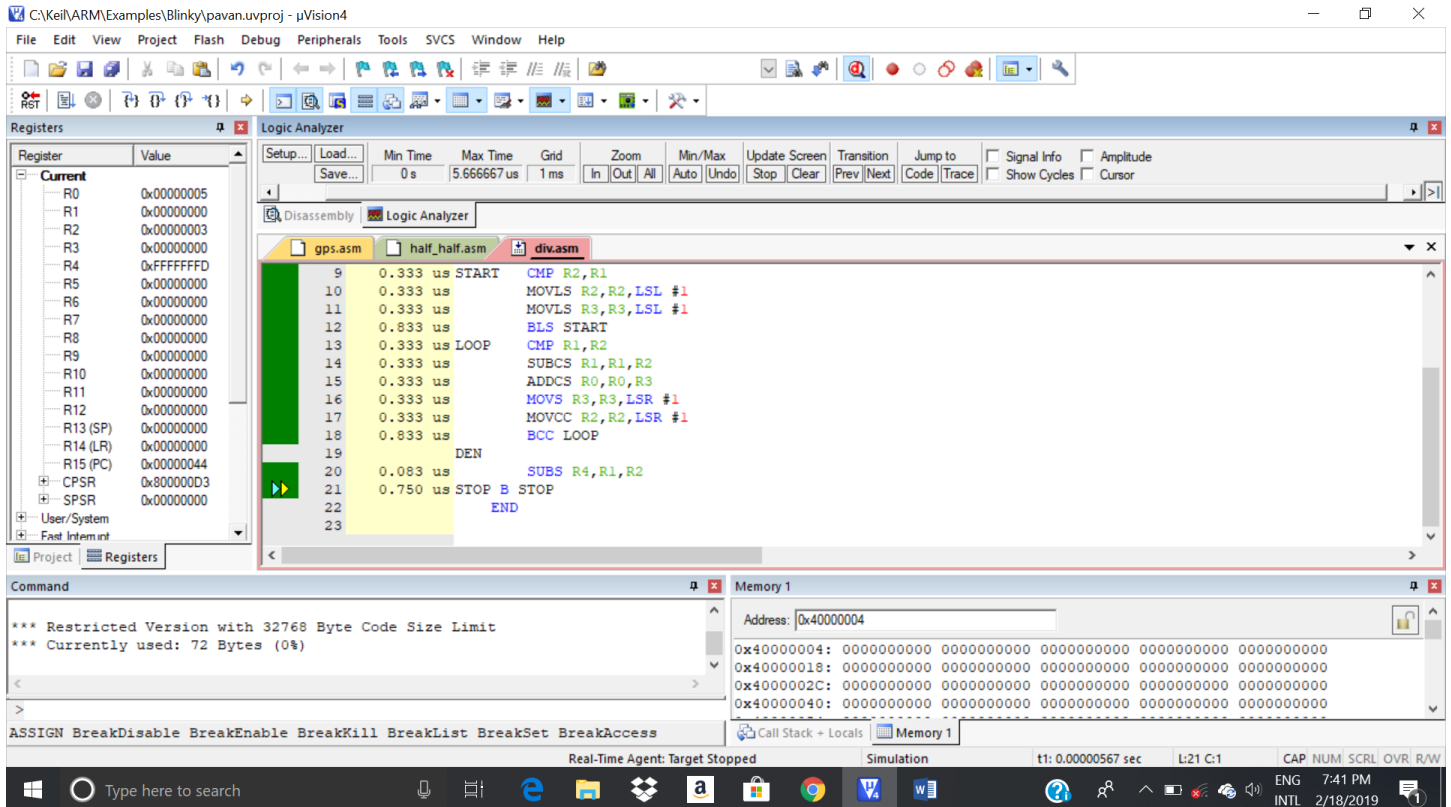
```

MOVLS R3, R3, LSL #1
BLS START
LOOP
CMP R1, R2
SUBCS R1, R1, R2
ADDCS R0, R0, R3
MOVS R3, R3, LSR #1
MOVCC R2, R2, LSR #1
BCC LOOP

DEN
SUBS R4, R1, R2

STOP B STOP
END

```



```

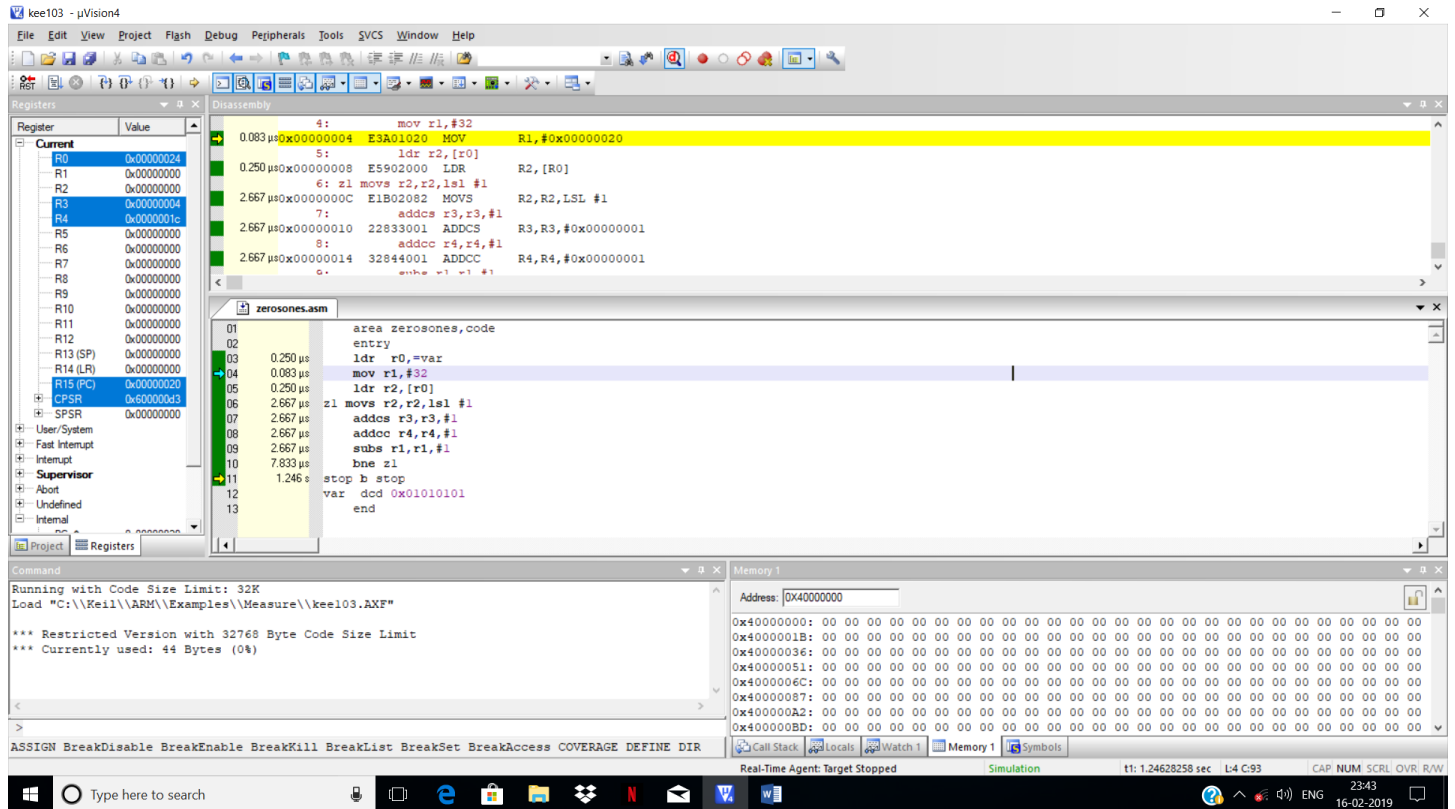
// ASSEMBLY PROGRAM TO COUNT NO OF ZEROS AND ONES
AREA ZEROS_ONES, CODE
ENTRY
LDR R0, =VAR
MOV R1, #32
LDR R2, [R0]
Z1 MOVS R2, R2, LSL #1
ADDCS R3, R3, #1
ADDC R4, R4, #1

```

```

SUBS R1,R1,#1
BNE Z1
STOP B STOP
VAR DCD 0X01010101
END

```



```

// PROGRAM TO FIND SUM OF NEGATIVE ELEMENTS IN ARRAY
CODE 1:

```

```

AREA NEGATIVESUM, CODE
ENTRY
LDR R0,=0X40000000
MOV R5,#05
MOV R6,#00
NEG LDR R2,[R0],#4
MOV R3,R2
MOVS R2,R2,LSL #1
ADDCS R6,R6,R3
SUBS R5,R5,#1
BNE NEG
STOP B STOP
END

```


CODE 2:

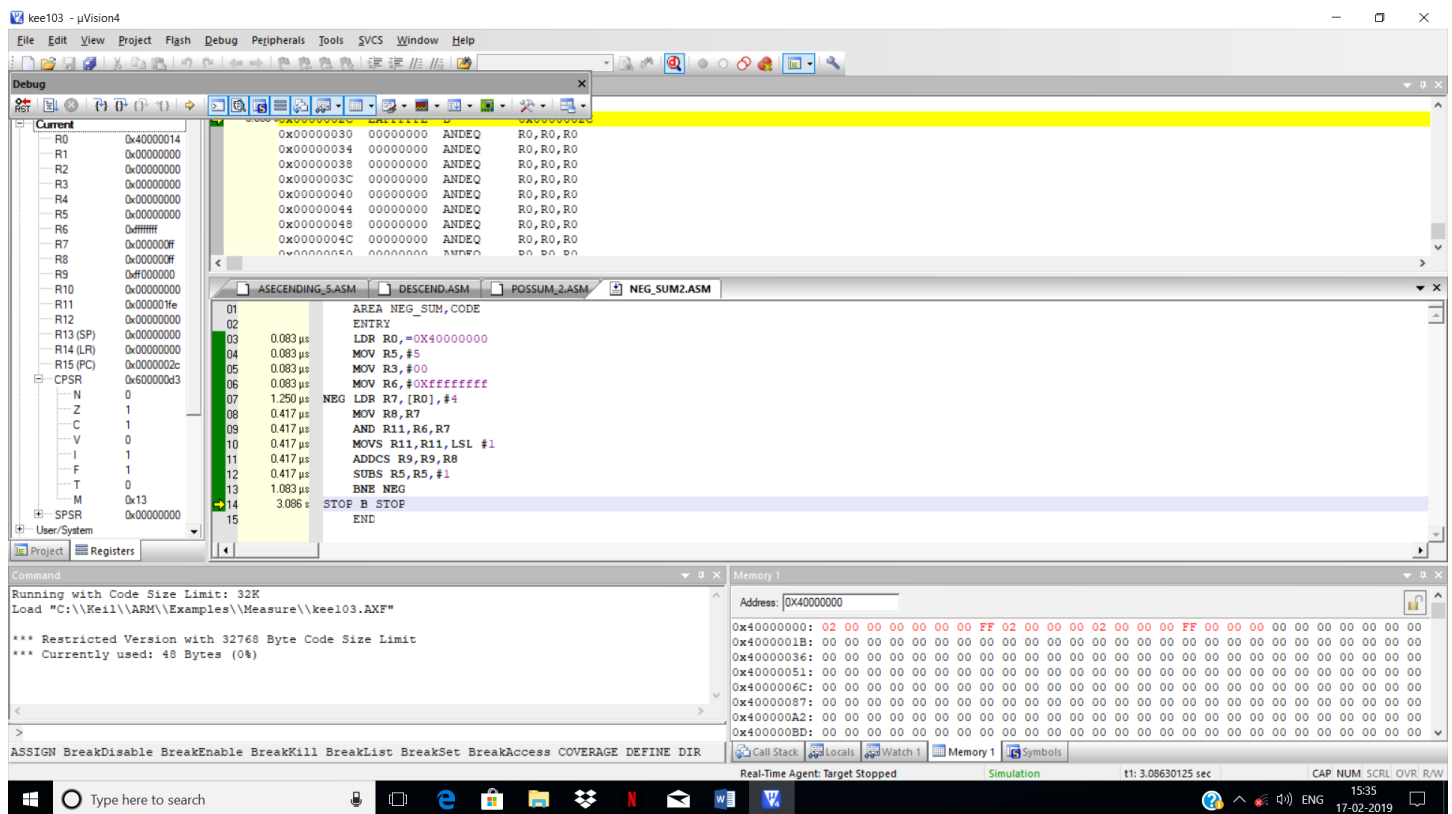
```

AREA NEG_SUM, CODE
ENTRY
LDR R0, =0X40000000
MOV R5, #5
MOV R3, #00
MOV R6, #0Xffffff
NEG LDR R7, [R0], #4
MOV R8, R7
AND R11, R6, R7
MOVS R11, R11, LSL #1
ADDCS R9, R9, R8
SUBS R5, R5, #1
BNE NEG

```

STOP B STOP

END



//PROGRAM TO FIND SUM OF POSITIVE NUMBERS IN ARRAY

CODE 1:

```

AREA POS_SUM, CODE
ENTRY
LDR R0, =0X40000000
MOV R5, #05
MOV R6, #00
POS LDR R2, [R0], #4
MOV R3, R2
MOVS R2, R2, LSL #1
ADDC R6, R6, R3
SUBS R5, R5, #1

```

```

BNE POS
STOP B STOP
END

```

Debug

Current

R0 0x00000014
R1 0x00000000
R2 0x00000010
R3 0x00000098
R4 0x00000000
R5 0x00000000
R6 0x0000001b
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x00000000
R15 (PC) 0x00000024
CPSR 0x60000043
N 0
Z 1
C 1
V 0
I 1
F 1
T 0
M 0x13
User/System 0x00000000

12: STOP B STOP

0.250 μs 0x00000024 EFFFFFFF B 0x00000024

0x00000028 00000000 ANDEQ R0,R0,R0
0x0000002C 00000000 ANDEQ R0,R0,R0
0x00000030 00000000 ANDEQ R0,R0,R0
0x00000034 00000000 ANDEQ R0,R0,R0
0x00000038 00000000 ANDEQ R0,R0,R0
0x0000003C 00000000 ANDEQ R0,R0,R0

zerosones.asm EVEN_SUM2.ASM ODD_SUM2.ASM Negative_Sum.ASM POSITIVESUM.ASM

01 AREA POS_SUM, CODE
02 ENTRY
03 0.083 μs LDR R0,=0x40000000
04 0.083 μs MOV R5,#5
05 0.083 μs MOV R6,#00
06 1.250 μs POS LDR R2,[R0],#4
07 0.417 μs MOV R3,R2
08 0.417 μs MOV R2,R2,LSL #1
09 0.417 μs ADDCC R6,R6,R3
10 0.417 μs SUBS R5,R5,#1
11 1.083 μs BNE POS
12 0.500 μs STOP B STOP
13 END

Command

Running with Code Size Limit: 32K
Load "C:\\Keil\\ARM\\Examples\\Measure\\keel03.AMF"

*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 40 Bytes (0%)

Memory 1

Address: 0x40000000

0x40000000: 00 00 00 00 FF 00 00 02 00 00 00 02 00 00 00 F8 00 00 02 00 00 00 00 00 00
0x4000001B: 00
0x40000036: 00
0x40000051: 00
0x4000006C: 00
0x40000087: 00
0x400000A2: 00
0x400000BD: 00

ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR

Real-Time Agent: Target Stopped Simulation t1: 0.0000475 sec CAP NUM SCRL OVR RAW

Type here to search

00:30 17-02-2019

```

CODE 2: AREA POS_SUM, CODE
        ENTRY
        LDR R0,=0x40000000
        MOV R5,#5
        MOV R3,#00
        MOV R6,#0xffffffff
        POS LDR R7,[R0],#4
        MOV R8,R7
        AND R11,R6,R7
        MOV R11,R11,LSL #1
        ADDCC R9,R9,R8
        SUBS R5,R5,#1
        BNE POS
        STOP B STOP
        END

```

keel03 - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Debug

Current

R0 0x40000014
R1 0x00000000
R2 0x00000000
R3 0x00000000
R4 0x00000000
R5 0x00000000
R6 0x00000000
R7 0x00000002
R8 0x00000002
R9 0x00000006
R10 0x00000000
R11 0x00000004
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x00000000
R15 (PC) 0x0000002c
CPSR 0x600000d3
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC \$ 0x0000002c
Mode Supervisor

Registers

Command

Running with Code Size Limit: 32K
Load "C:\\Keil\\ARM\\Examples\\Measure\\keel03.AXF"
*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 48 Bytes (0%)

Memory 1

Address: 0x40000000

0x40000000: 02 00 00 00 00 00 00 00 FF 02 00 00 00 00 00 00 00 F8 02 00 00 00 00 00 00 00 00 00 00 00 00
0x40000001B: 00
0x400000036: 00
0x400000051: 00
0x40000006C: 00
0x400000087: 00
0x4000000A2: 00
0x4000000BD: 00

Real-Time Agent: Target Stopped

Simulation

t1: 0.78015175 sec L2: C121 CAP: NUM SCRL OVR: R/W

15:28
17-02-2019

//ASSEMBLY PROGRAM TO ARRANGE THE NUMBERS IN ASCENDING ORDER

CODE 1:

AREA ASECEND.CODE

ENTRY

MOV R0,#5

OUT LDR R5,=0X40000000

ADD R6,R5,#1

MOV R3,#4

IN LDRB R1,[R5]

LDRB R2,[R6]

CMP R1,R2

BCC LOP

MOV R4,R2

MOV R2,R1

MOV R1,R4

LOP STRB R1,[R5]

STRB R2,[R6]

ADD R5,R5,#1

ADD R6,R6,#1

SUBS R3,R3,#1

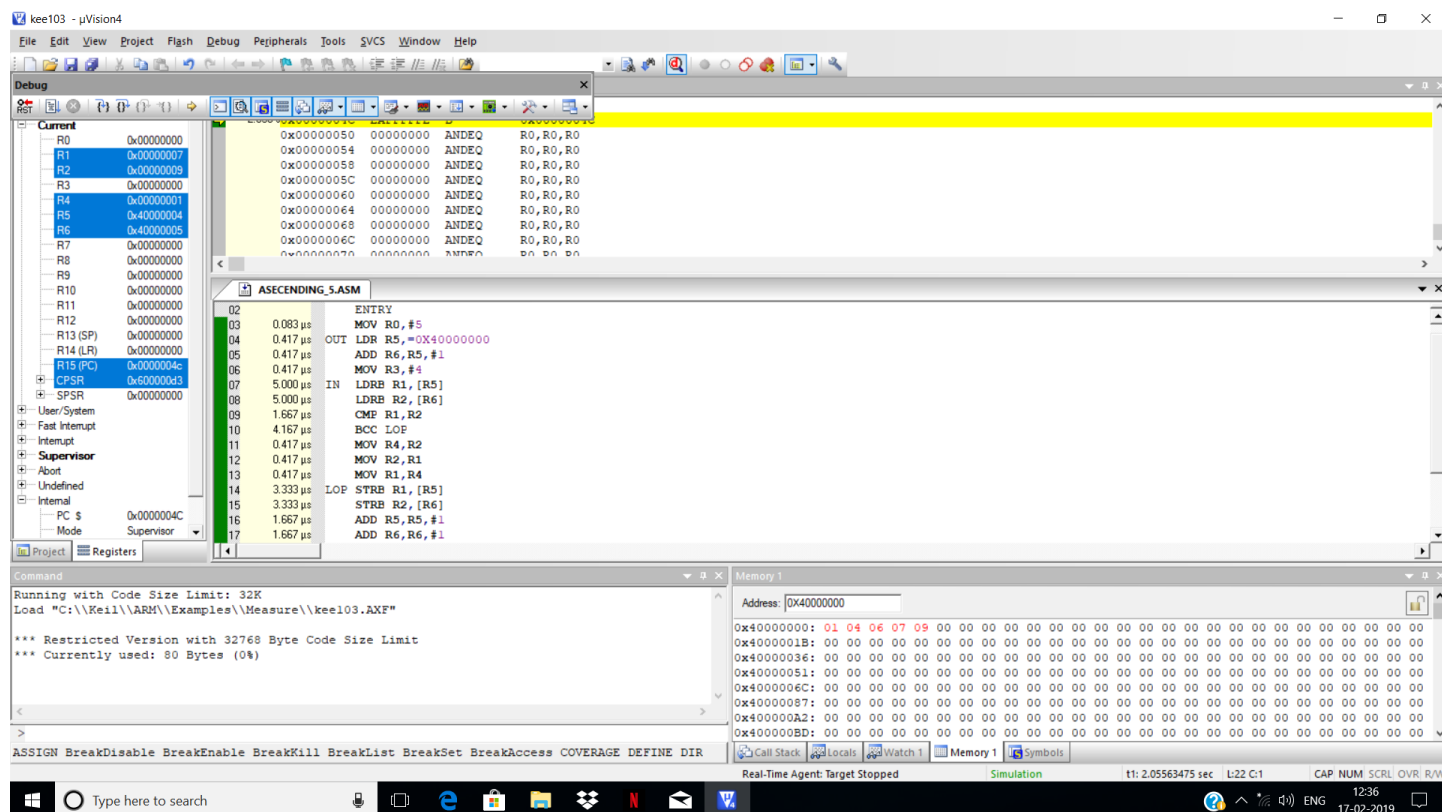
BNE IN

SUBS R0,R0,#1

BNE OUT

STOP B STOP

END



CODE 2:

AREA A0.CODE

ENTRY

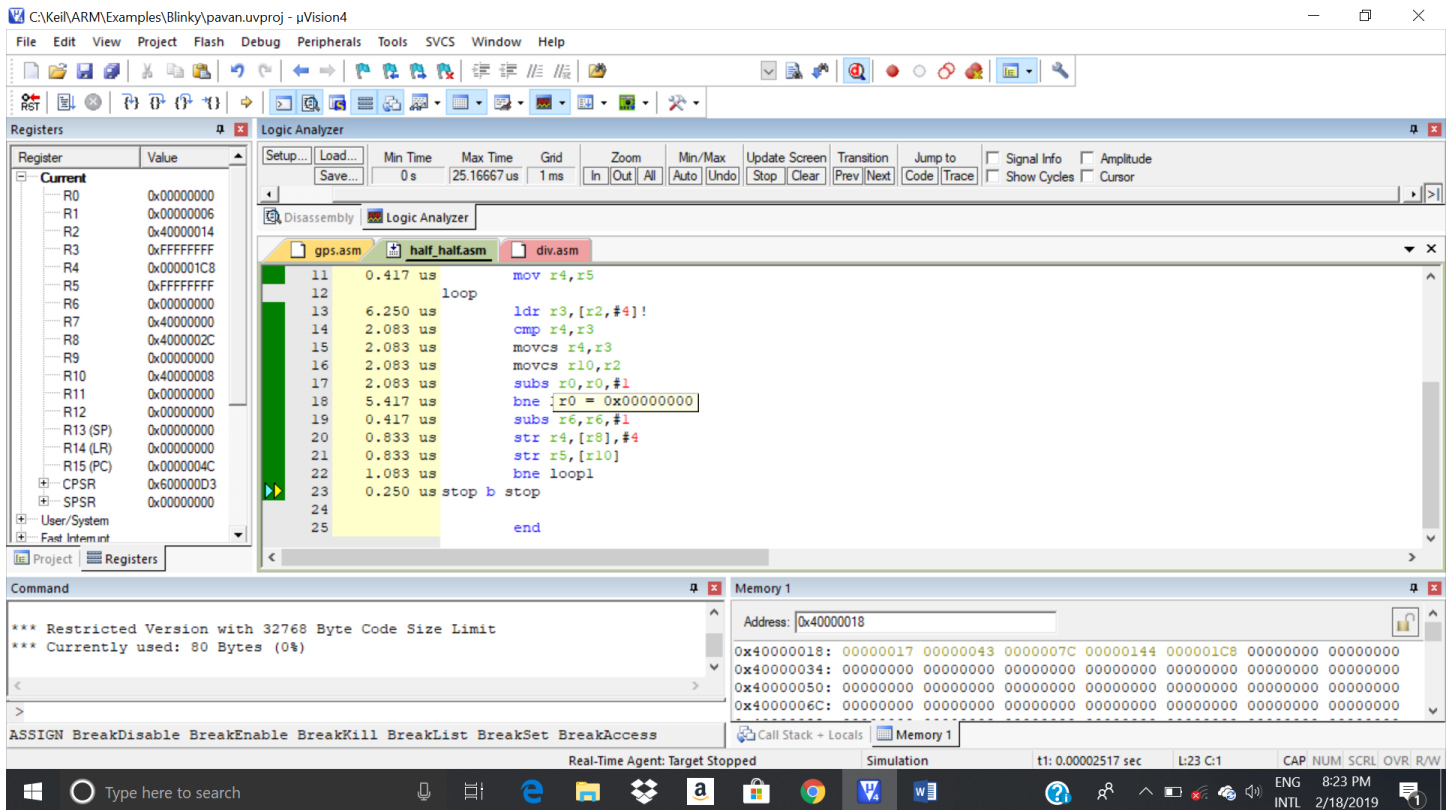
```

MOV R6,#5
MOV R0,#0
MOV R1,#6
MOV R5,#0xFFFFFFFF
LDR R7,=0x40000000
LDR R8,=0x40000018
LOOP1
SUB R0,R1,#1
MOV R2,R7
MOV R4,R5

LOOP
LDR R3,[R2,#4]!
CMP R4,R3
MOVCS R4,R3
MOVCS R10,R2
SUBS R0,R0,#1
BNE LOOP
SUBS R6,R6,#1
STR R4,[R8],#4
STR R5,[R10]
BNE LOOP1

STOP B STOP
END

```



// ASSEMBLY PROGRAM TO ARRANGE NUMBERS IN DESCENDING

CODE 1: AREA DESCEND, CODE

ENTRY

MOV R0,#5

OUT LDR R5,=0x40000000

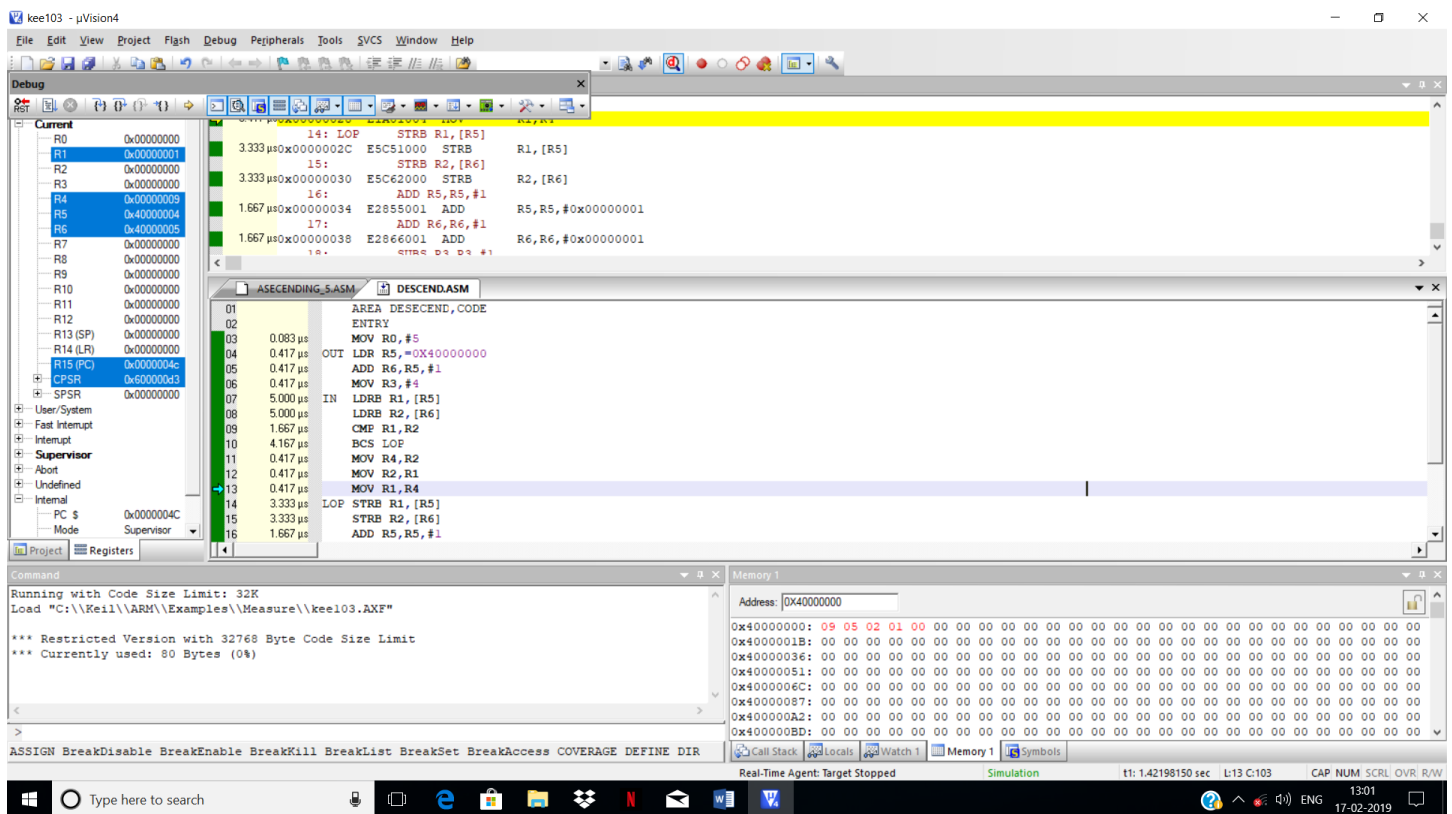
ADD R6,R5,#1

MOV R3,#4

```

IN  LDRB R1,[R5]
    LDRB R2,[R6]
    CMP R1,R2
    BCS LOP
    MOV R4,R2
    MOV R2,R1
    MOV R1,R4
LOP  STRB R1,[R5]
    STRB R2,[R6]
    ADD R5,R5,#1
    ADD R6,R6,#1
    SUBS R3,R3,#1
    BNE IN
    SUBS R0,R0,#1
    BNE OUT
STOP B STOP
END

```



```

CODE 2: AREA AO.CODE
      ENTRY
      MOV R6,#5
      MOV R0,#0
      MOV R1,#6
      MOV R5,#0
      LDR R7,=0X40000000
      LDR R8,=0X40000018
LOOP1 SUB R0,R1,#1
      MOV R2,R7

```

```

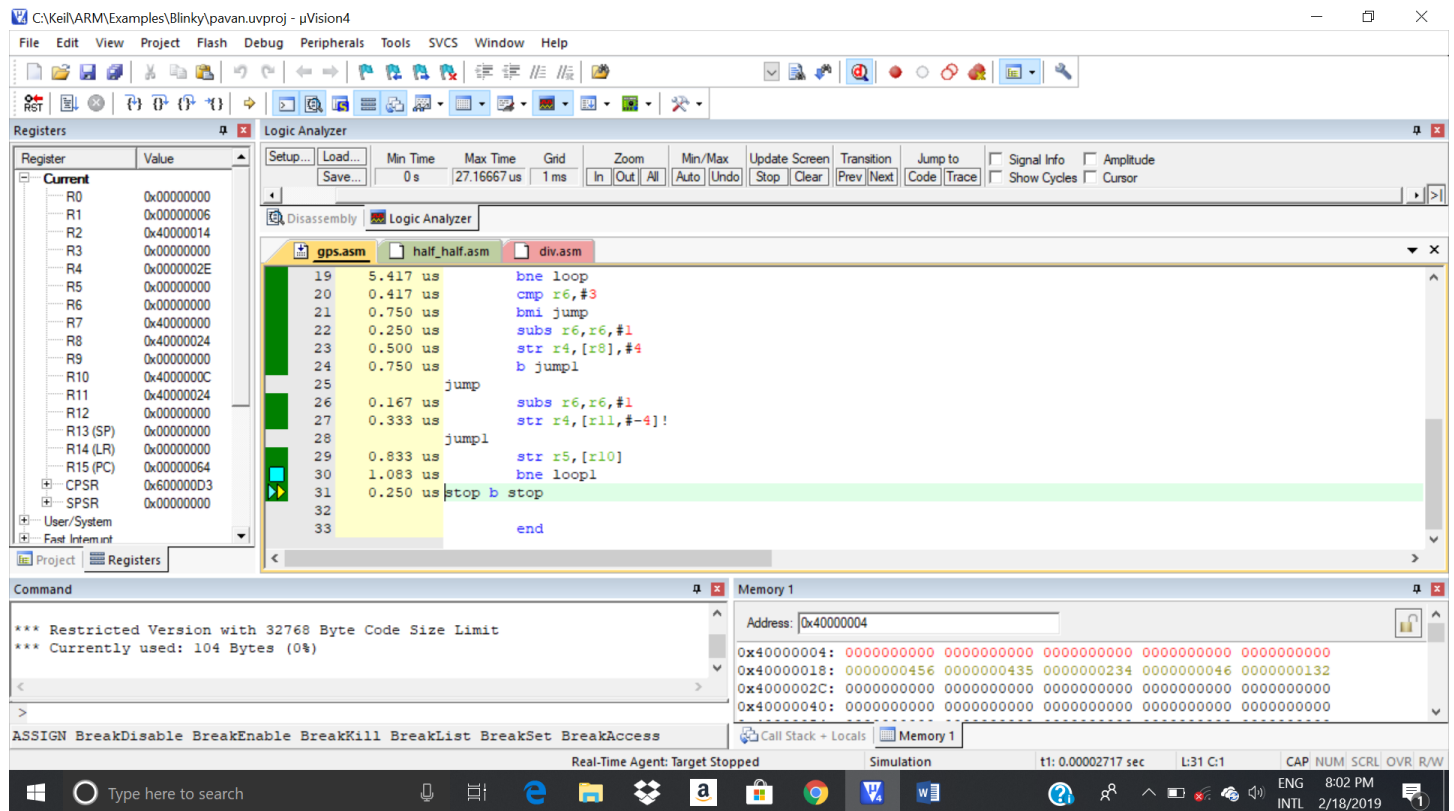
MOV R4,R5
LOOP
    LDR R3,[R2,#4]!
    CMP R4,R3
    MOVCC R4,R3
    MOVCC R10,R2
    SUBS R0,R0,#1
    BNE LOOP
    SUBS R6,R6,#1
    STR R4,[R8],#4
    STR R5,[R10]
    BNE LOOP1

```

```

STOP B STOP
END

```



```

// ASSEMBLY PROGRAM TO ARRANGE THE NUMBERS IN HALF ASCENDING AND HALF DESCENDING ORDER
CODE 1:

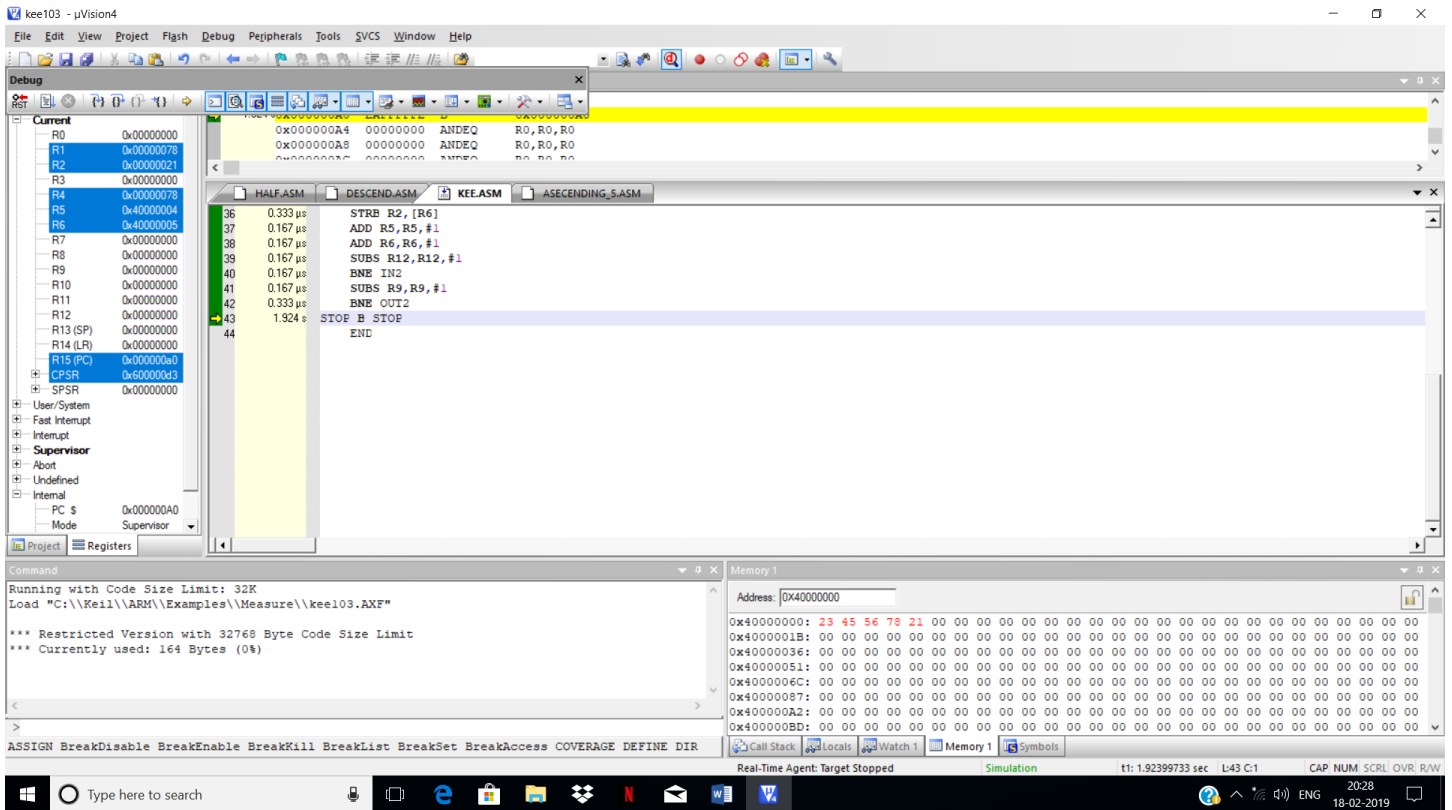
```

```

AREA HALF.CODE
ENTRY
MOV R0,#3
MOV R9,#2
MOV R10,#0
OUT1 LDR R5,=0X40000000

```

```
        ADD R6,R5,#1
        MOV R3,#2
IN1  LDRB R1,[R5]
        LDRB R2,[R6]
        CMP R1,R2
        BCC LOP
        MOV R4,R2
        MOV R2,R1
        MOV R1,R4
LOP   STRB R1,[R5]
        STRB R2,[R6]
        ADD R5,R5,#1
        ADD R6,R6,#1
        SUBS R3,R3,#1
        BNE IN1
        SUBS R0,R0,#1
        BNE OUT1
        BEQ OUT2
OUT2  LDR R5,=0X40000003
        ADD R6,R5,#1
        MOV R12,#1
IN2  LDRB R1,[R5]
        LDRB R2,[R6]
        CMP R1,R2
        BCS LOP2
        MOV R4,R2
        MOV R2,R1
        MOV R1,R4
LOP2  STRB R1,[R5]
        STRB R2,[R6]
        ADD R5,R5,#1
        ADD R6,R6,#1
        SUBS R12,R12,#1
        BNE IN2
        SUBS R9,R9,#1
        BNE OUT2
STOP B STOP
        END
```



CODE 2: AREA A0, CODE

ENTRY

MOV R6, #5

MOV R0, #0

MOV R1, #6

MOV R5, #0

LDR R7, =0X40000000

LDR R8, =0X40000018

LDR R11, =0X4000002C

LOOP1 SUB R0, R1, #1

MOV R2, R7

MOV R4, R5

LOOP

LDR R3, [R2, #4]!

CMP R4, R3

MOVCC R4, R3

MOVCC R10, R2

SUBS R0, R0, #1

BNE LOOP

CMP R6, #3

BMI JUMP

SUBS R6, R6, #1

STR R4, [R8], #4

B JUMP1

JUMP

SUBS R6, R6, #1

STR R4, [R11, #-4]!

JUMP1

```
STR R5,[R10]
```

```
BNE LOOP1
```

```
STOP B STOP
```

```
END
```

CA\Keil\ARM\Examples\Blinky\pavan.uvproj - µVision4

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

Registers

Register	Value
R0	0x00000000
R1	0x00000006
R2	0x40000014
R3	0x00000000
R4	0x0000002E
R5	0x00000000
R6	0x00000000
R7	0x40000000
R8	0x40000024
R9	0x00000000
R10	0x4000000C
R11	0x40000024
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000064
CPSR	0x600000D3
SPSR	0x00000000
User/System	
Fast Interrupt	

Logic Analyzer

Setup... Load... Min Time Max Time Grid Zoom Min/Max Update Screen Transition Jump to Signal Info Amplitude
Save... 0 s 27.16667 us 1 ms In Out All Auto Undo Stop Clear Prev Next Code Trace Show Cycles Cursor

Disassembly Logic Analyzer

gps.asm half_half.asm div.asm

```

19 5.417 us bne loop
20 0.417 us cmp r6,#3
21 0.750 us bmi jump
22 0.250 us subs r6,r6,#1
23 0.500 us str r4,[r8],#4
24 0.750 us b jump1
25
26 0.167 us jump subs r6,r6,#1
27 0.333 us str r4,[r11],#-4!
28
29 0.833 us jump1
30 1.083 us str r5,[r10]
31 0.250 us bne loop1
32 stop b stop
33 end

```

Command

```

*** Restricted Version with 32768 Byte Code Size Limit
*** Currently used: 104 Bytes (0%)

```

Memory 1

Address: 0x40000004

```

0x40000004: 00000000 00000000 00000000 00000000 00000000
0x40000018: 00000046 00000043 00000023 00000046 00000013
0x4000002C: 00000000 00000000 00000000 00000000 00000000
0x40000040: 00000000 00000000 00000000 00000000 00000000

```

Call Stack + Locals Memory 1

Real-Time Agent: Target Stopped Simulation tt: 0.00002717 sec L:31 C:1 CAP NUM SCRL OVR: R/W

ENG 8:02 PM
INTL 2/18/2019

