Microprocessor & Computer Architecture (µpCA)

UE19CS252

Dr. D. C. Kiran

Department of
Computer Science and Engineering

Microprocessor & Computer Architecture (μpCA)

Block Transfer Instructions: Stack

Dr. D. C. Kiran

Department of Computer Science and Engineering

Microprocessor & Computer Architecture (μpCA)

Syllabus

Unit 1: Basic Processor Architecture and Design

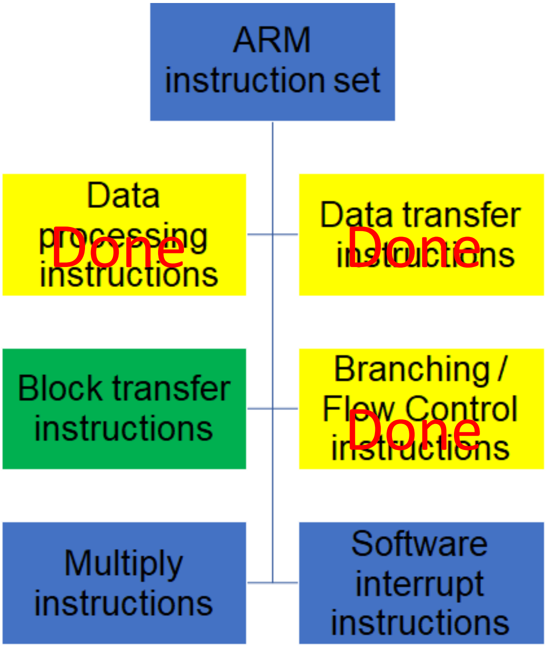
- ~~Microprocessor Overview~~
- ~~CISC VS RISC~~
- ~~Introduction to ARM Processor & Applications~~
- ~~ARM Architecture Overview~~
- ~~Different ARM processor Modes~~
- ~~Register Bank~~
- ~~ARM Program structure~~
- ~~ARM Instruction Format~~
- **ARM INSTRUCTION SET**

  ~~Data Processing Instructions~~

  ~~Flow Control Instructions~~

  ~~Data Transfer Instructions~~

  Block Transfer Instructions (Stack)


ARM instruction set
Data processing instructions
Done
Data transfer instructions
Done
Block transfer instructions
Branching / Flow Control instructions
Done
Multiply instructions
Software interrupt instructions

Block Transfer Instructions (Stack)

The Memory access can be in FILO fashion.

i.e Can be treated like STACK.

R13 is a stack pointer which will keep the address of TOP of the STACK.

Mainly used in Procedural Call.

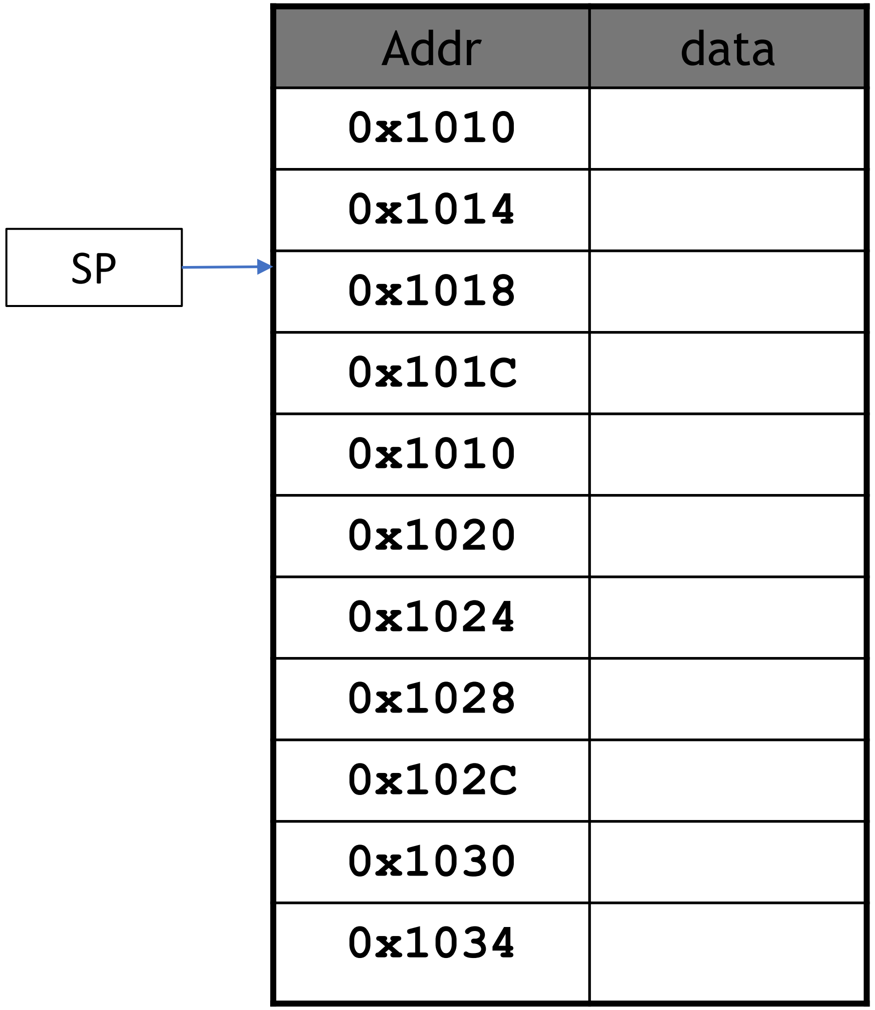
Stack can grow upward or downward direction based on the MODE used by the user in the program.

Addr	data
0x1010	
0x1014	
0x1018	10
0x101C	20
0x1010	30
0x1020	40
0x1024	50
0x1028	60
0x102C	
0x1030	
0x1034	

Block Transfer Instructions (Stack)

10 20 10 140 15 60 27

R13=0x1018



SP

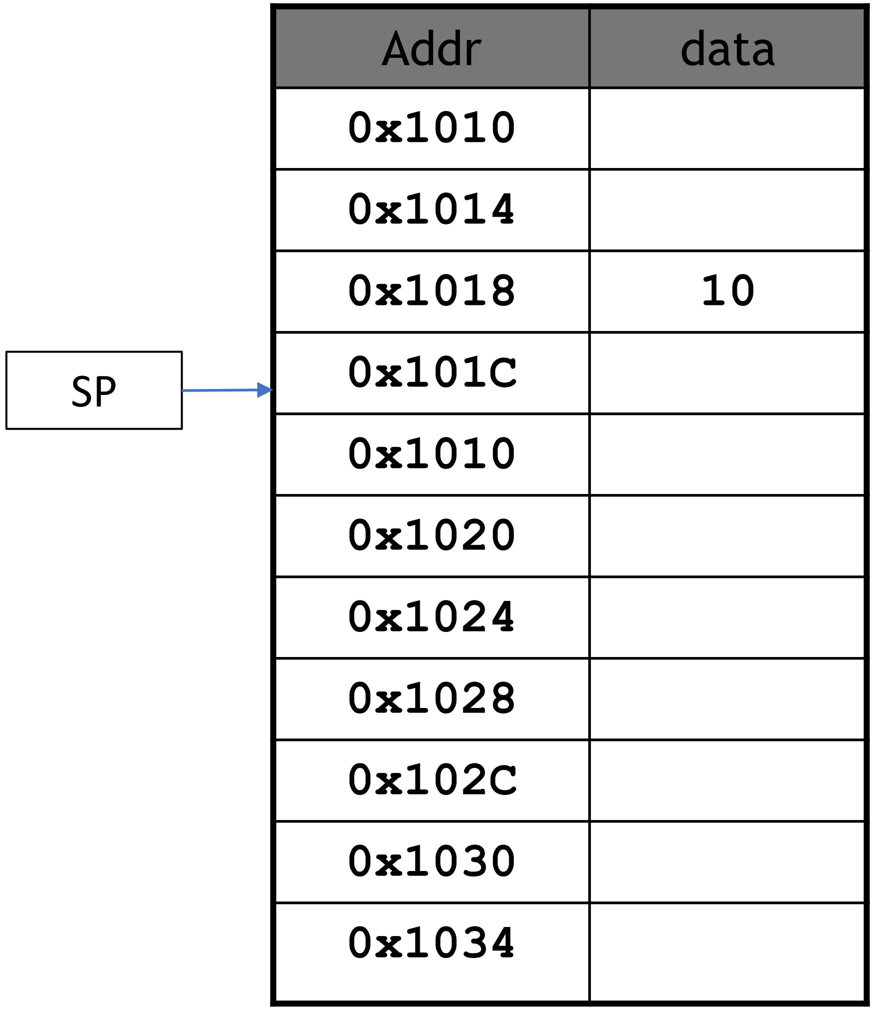
Addr	data
0x1010	
0x1014	
0x1018	
0x101C	
0x1010	
0x1020	
0x1024	
0x1028	
0x102C	
0x1030	
0x1034	

Microprocessor & Computer Architecture (μpCA)

Block Transfer Instructions (Stack)

10 20 75 140 15 60 27

R13=0x101C



SP

Addr	data
0x1010	
0x1014	
0x1018	10
0x101C	
0x1010	
0x1020	
0x1024	
0x1028	
0x102C	
0x1030	
0x1034	

Microprocessor & Computer Architecture (μpCA)

Block Transfer Instructions (Stack)

10 20 75 140 15 60 27

R13=0x1010

SP

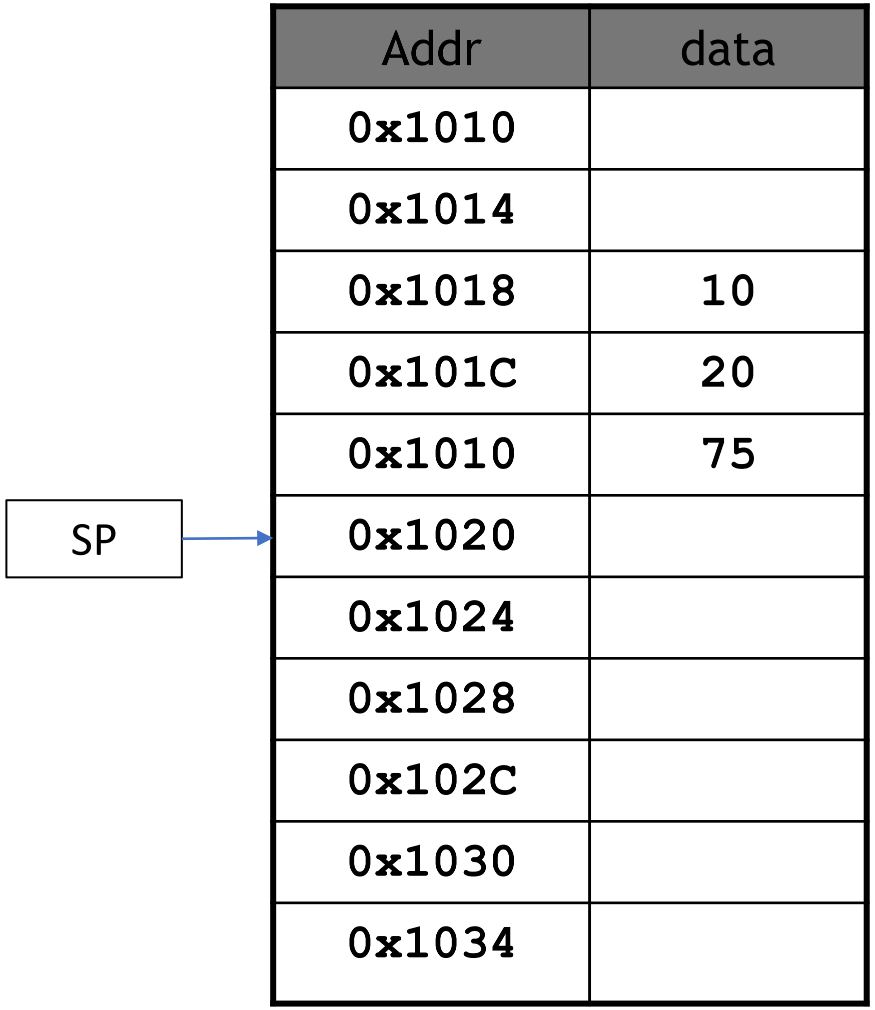
Addr	data
0x1010	
0x1014	
0x1018	10
0x101C	20
0x1010	
0x1020	
0x1024	
0x1028	
0x102C	
0x1030	
0x1034	

Microprocessor & Computer Architecture (μpCA)

Block Transfer Instructions (Stack)

10 20 75 140 15 60 27

R13=0x1020



SP

Addr	data
0x1010	
0x1014	
0x1018	10
0x101C	20
0x1010	75
0x1020	
0x1024	
0x1028	
0x102C	
0x1030	
0x1034	

Block Transfer Instructions (Stack)

10 20 75 140 15 60 27

R13=0x1024

Addr	data
0x1010	
0x1014	
0x1018	10
0x101C	20
0x1010	75
0x1020	140
0x1024	
0x1028	
0x102C	
0x1030	
0x1034	

SP → 0x1024

Block Transfer Instructions (Stack)

Syntax:

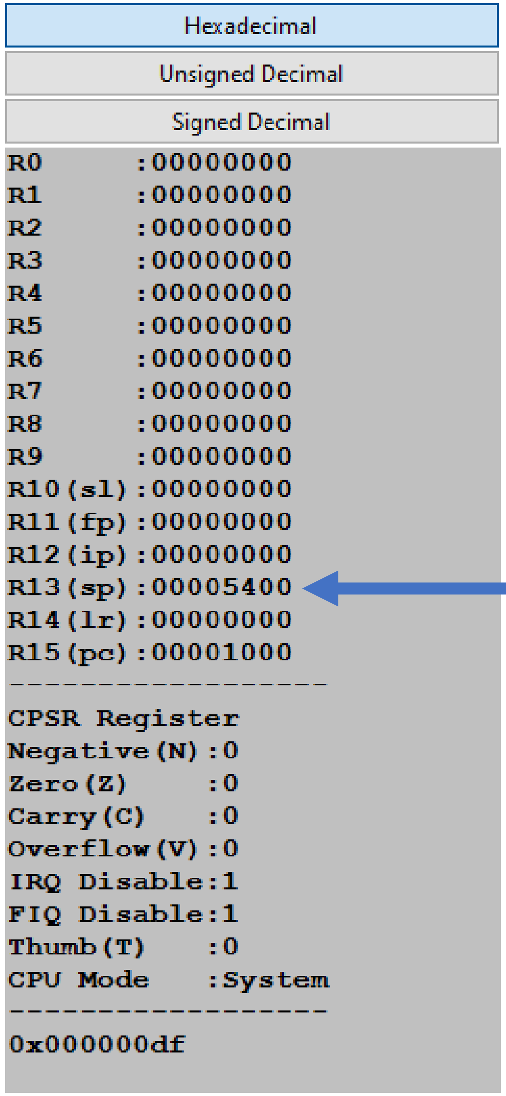
<LDM/STM> <Addressing Mode>R13 {!},Registers

Addressing Mode	=LDM	=STM
Full ascending (**FA**)	**LDMDA**	**STMIB**
Full descending (**FD**)	**LDMIA**	**STMDB**
Empty ascending (**EA**)	**LDMDB**	**STMIA**
Empty descending (**ED**)	**LDMIB**	**STMDA**

STM is used to PUSH on to the STACK
LDM is used to POP from the STACK

Microprocessor & Computer Architecture (μpCA)

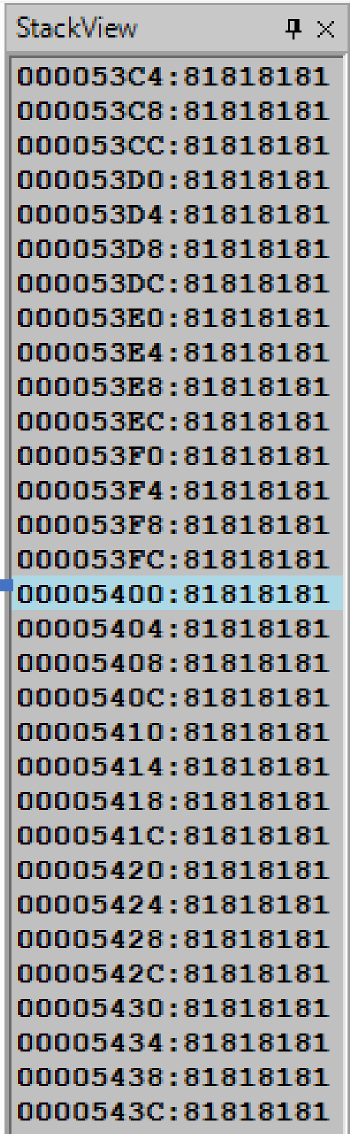
Block Transfer Instructions (Stack)



```
R0       :00000000
R1       :00000000
R2       :00000000
R3       :00000000
R4       :00000000
R5       :00000000
R6       :00000000
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):00001000
------------------
CPSR Register
Negative(N):0
Zero(Z)    :0
Carry(C)   :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :0
CPU Mode   :System
------------------
0x000000df
```

```
.text
MOV R0,#4
MOV R1,#5
STMEA R13!, {R0,R1}
.end
```

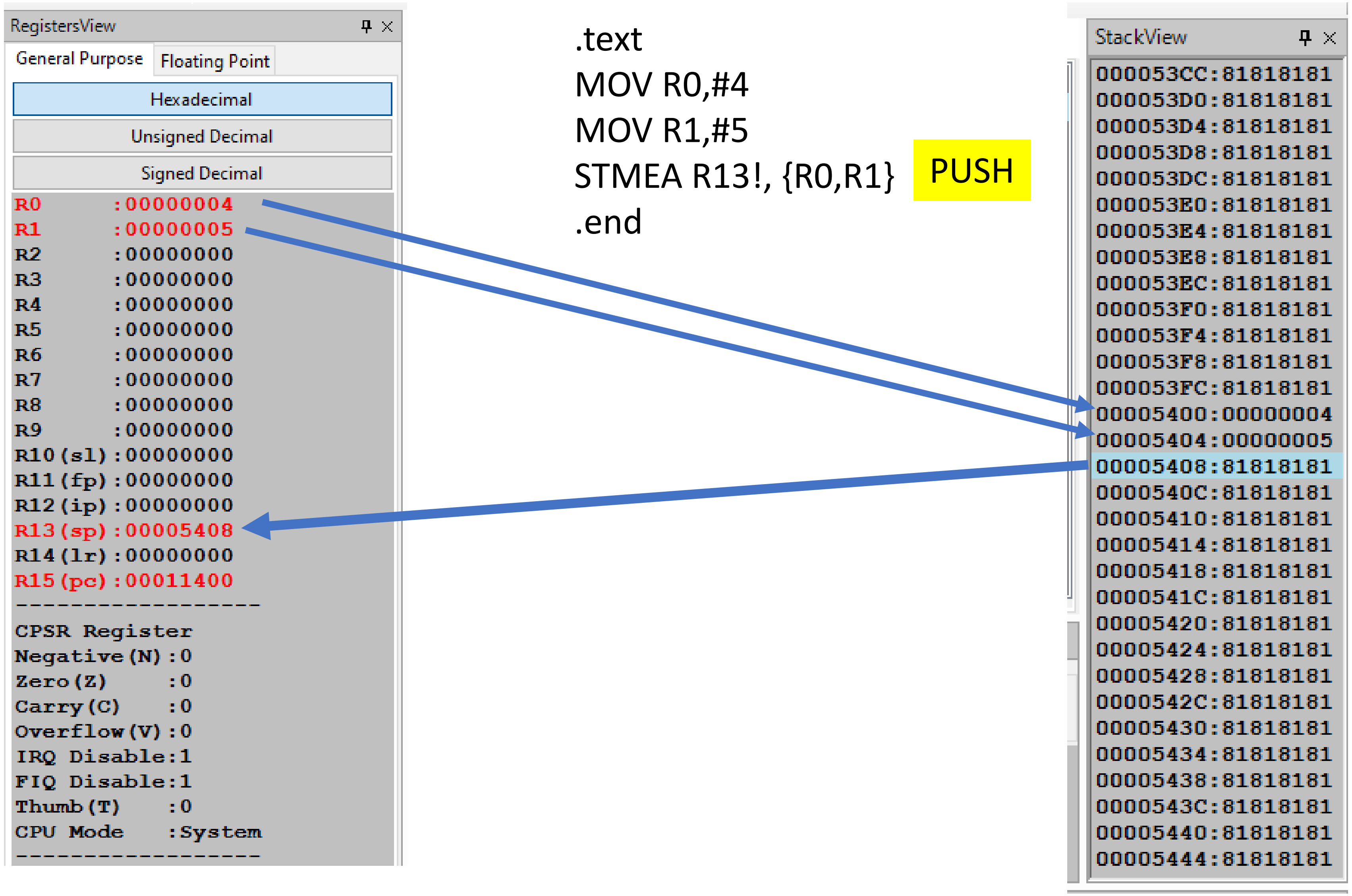
Initially Stack is Empty



```
000053C4:81818181
000053C8:81818181
000053CC:81818181
000053D0:81818181
000053D4:81818181
000053D8:81818181
000053DC:81818181
000053E0:81818181
000053E4:81818181
000053E8:81818181
000053EC:81818181
000053F0:81818181
000053F4:81818181
000053F8:81818181
000053FC:81818181
00005400:81818181
00005404:81818181
00005408:81818181
0000540C:81818181
00005410:81818181
00005414:81818181
00005418:81818181
0000541C:81818181
00005420:81818181
00005424:81818181
00005428:81818181
0000542C:81818181
00005430:81818181
00005434:81818181
00005438:81818181
0000543C:81818181
```

Microprocessor & Computer Architecture (μpCA)

Block Transfer Instructions (Stack)



```
.text
MOV R0,#4
MOV R1,#5
STMEA R13!, {R0,R1}
.end
```

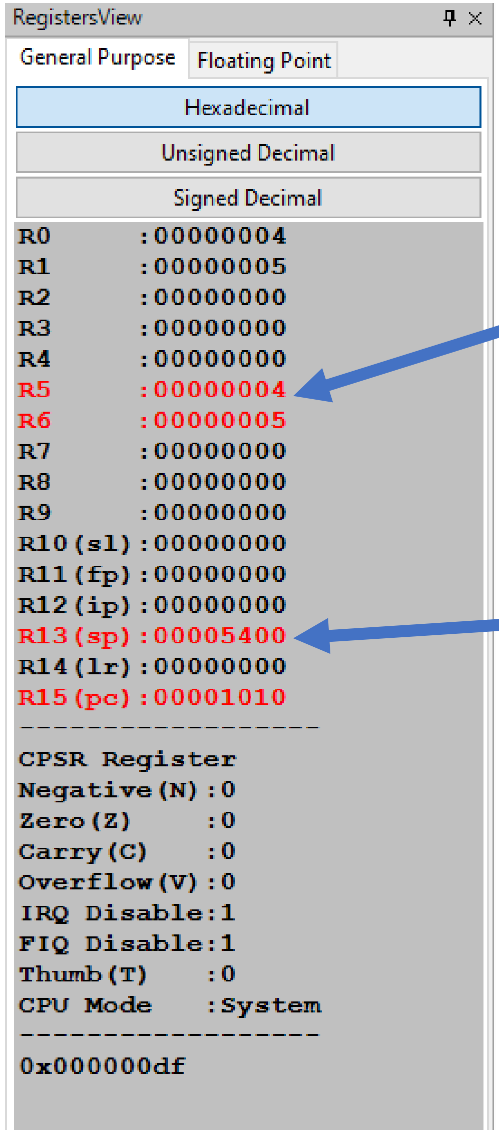
PUSH

```
R0       :00000004
R1       :00000005
R2       :00000000
R3       :00000000
R4       :00000000
R5       :00000000
R6       :00000000
R7       :00000000
R8       :00000000
R9       :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005408
R14(lr):00000000
R15(pc):00011400
------------------
CPSR Register
Negative(N):0
Zero(Z)     :0
Carry(C)    :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)    :0
CPU Mode    :System
------------------
```

```
000053CC:81818181
000053D0:81818181
000053D4:81818181
000053D8:81818181
000053DC:81818181
000053E0:81818181
000053E4:81818181
000053E8:81818181
000053EC:81818181
000053F0:81818181
000053F4:81818181
000053F8:81818181
000053FC:81818181
00005400:00000004
00005404:00000005
00005408:81818181
0000540C:81818181
00005410:81818181
00005414:81818181
00005418:81818181
0000541C:81818181
00005420:81818181
00005424:81818181
00005428:81818181
0000542C:81818181
00005430:81818181
00005434:81818181
00005438:81818181
0000543C:81818181
00005440:81818181
00005444:81818181
```

Microprocessor & Computer Architecture (μpCA)

Block Transfer Instructions (Stack)

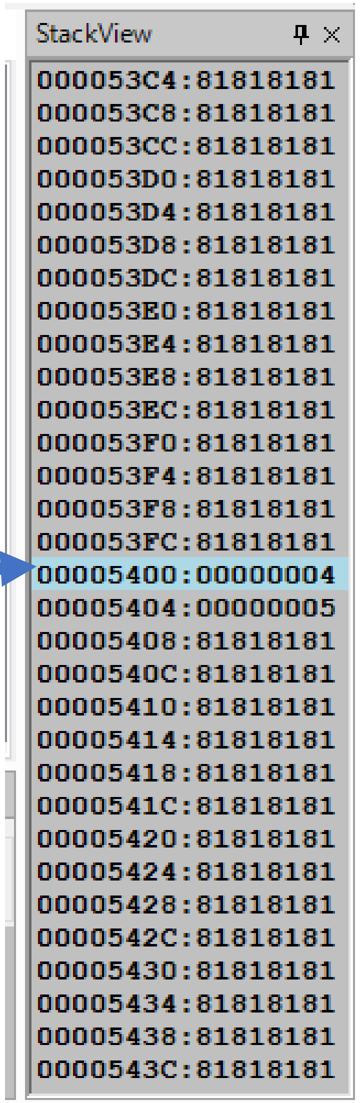


```
R0      :00000004
R1      :00000005
R2      :00000000
R3      :00000000
R4      :00000000
R5      :00000004
R6      :00000005
R7      :00000000
R8      :00000000
R9      :00000000
R10(sl):00000000
R11(fp):00000000
R12(ip):00000000
R13(sp):00005400
R14(lr):00000000
R15(pc):00001010
------------------
CPSR Register
Negative(N):0
Zero(Z)    :0
Carry(C)   :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :0
CPU Mode   :System
------------------
0x000000df
```

```
.text
MOV R0,#4
MOV R1,#5
STMEA R13!, {R0,R1}
LDMEA R13!,{R5,R6}
.end
```

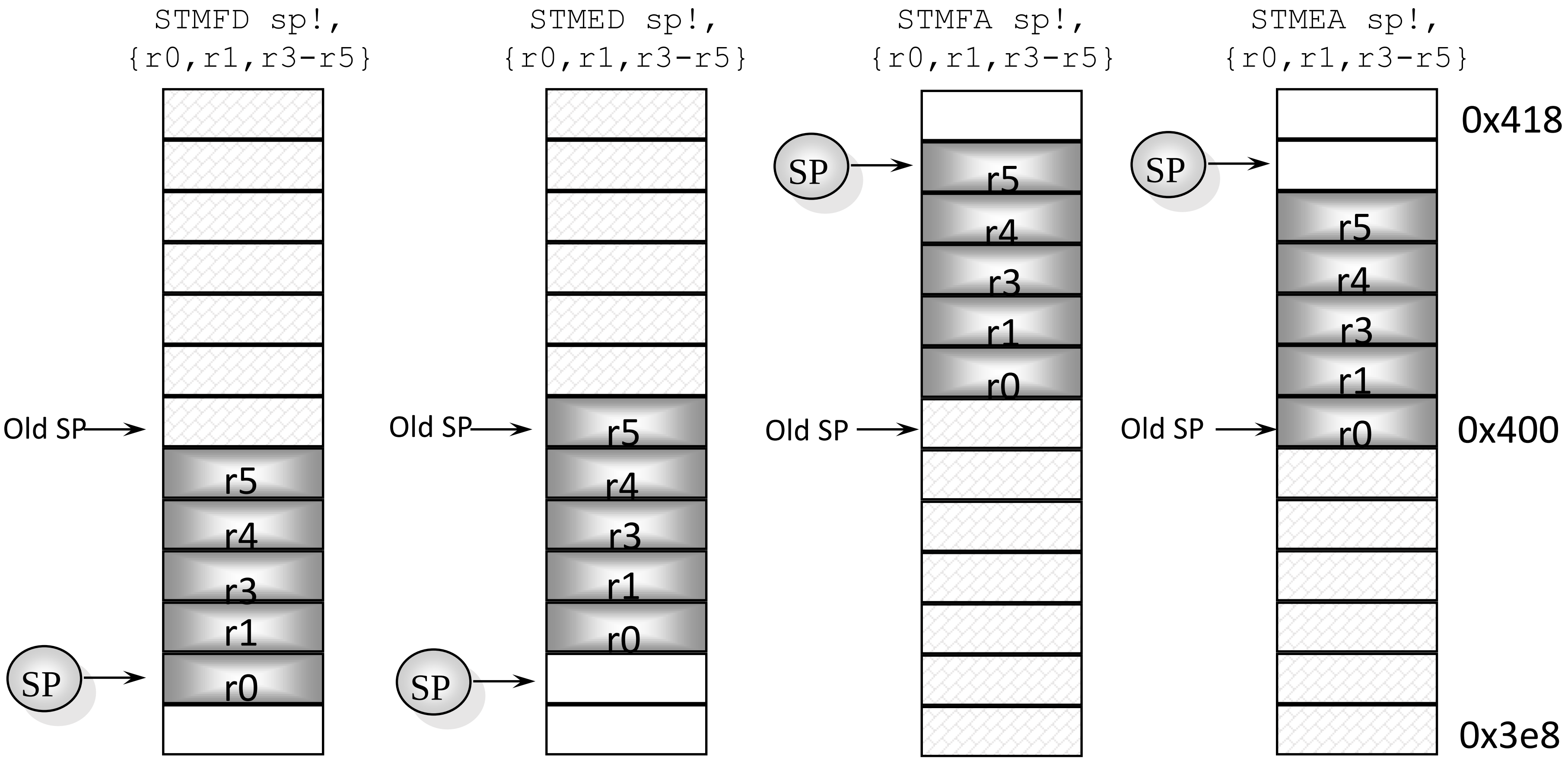
POP

Again the Stack is Empty

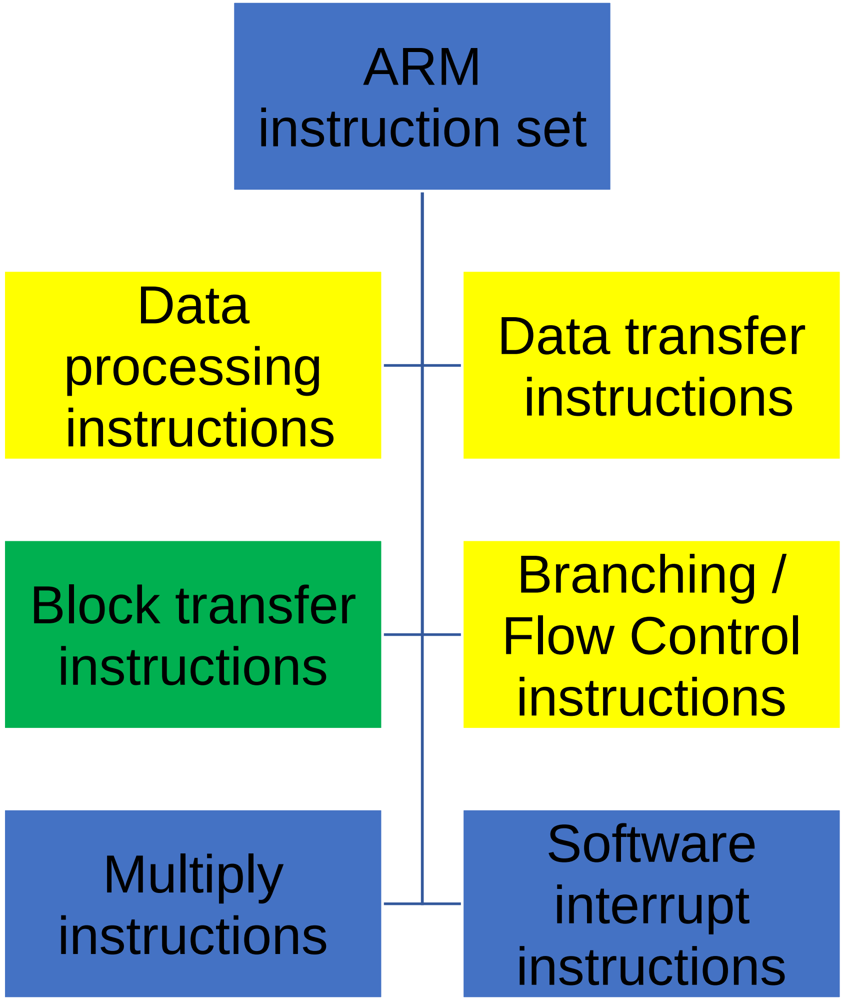


```
000053C4:81818181
000053C8:81818181
000053CC:81818181
000053D0:81818181
000053D4:81818181
000053D8:81818181
000053DC:81818181
000053E0:81818181
000053E4:81818181
000053E8:81818181
000053EC:81818181
000053F0:81818181
000053F4:81818181
000053F8:81818181
000053FC:81818181
00005400:00000004
00005404:00000005
00005408:81818181
0000540C:81818181
00005410:81818181
00005414:81818181
00005418:81818181
0000541C:81818181
00005420:81818181
00005424:81818181
00005428:81818181
0000542C:81818181
00005430:81818181
00005434:81818181
00005438:81818181
0000543C:81818181
```

Stack Examples



NEXT Session : Procedure Call


ARM instruction set
Data processing instructions
Data transfer instructions
Block transfer instructions
Branching / Flow Control instructions
Multiply instructions
Software interrupt instructions

PES UNIVERSITY ONLINE

THANK YOU

Dr. D. C. Kiran

Department of Computer Science and Engineering

dckiran@pes.edu

9829935135