Microprocessor & Computer Architecture (µpCA)

UE19CS252

Dr. D. C. Kiran

Department of
Computer Science and Engineering

PES UNIVERSITY ONLINE

Microprocessor & Computer Architecture (µpCA)

ARM Program Structure & Instruction Format

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

Microprocessor & Computer Architecture (µpCA)

ARM Program Structure

Address	Instruction & Data
	.text
Address of Instruction 1	ARM Instruction_1
Address of Instruction 2	ARM Instruction_3
Address of Instruction 3	ARM Instruction_3
…….	…….
…….	…….
Address of Instruction n	ARM Instruction_n
	.data
Address of Data1	Declaration of variable 1
Address of Data2	Declaration of variable 2
…….	…….
…….	…….
Address of Data n	Declaration of variable n

Note: Blue color depict the code written by the programmer
Red color depict the address assigned during execution

Microprocessor & Computer Architecture (µpCA)

ARM Program Structure

Address	Instruction	Meaning
	.text	
00001000:EF9F0014	LDR R0, =a	Load the Address of a to R0
00001004:EF9F1014	LDR R1,=b	Load the Address of b to R1
00001008:EF9F3014	LDR R3, =c	Load the Address of c to R3
0000100C:E5D14000	LDR R4, [r1]	Load the value (100) to R4
00001010:E5D05000	LDR R5, [r0]	Load the value (200) to R5
00001014:E0846005	Add R6, R4, R5	Add R4 & R5
00001018:E00360B0	STR R6, [r3]	Store the result(300) in the address specified in R3
	.data	
00001028:	a: .word 100	Variable ***a*** of data type ***word***
00001029:	b: word 200	Variable ***b*** of data type ***word***
0000102A:	c: word 0	Variable ***c*** of data type ***word***

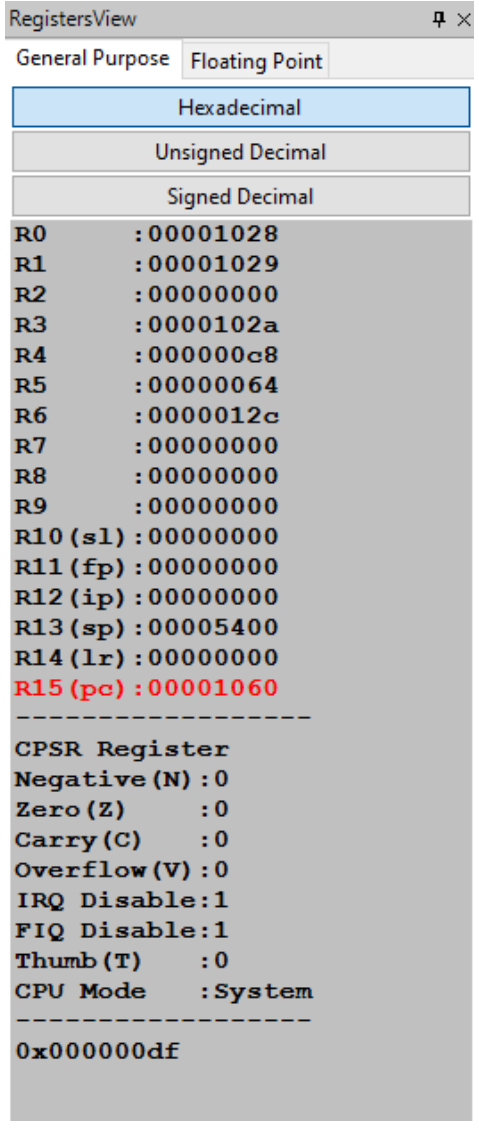
Note: the datatype may be byte, half word, asciz

Microprocessor & Computer Architecture (µpCA)

Register View (ARMSIM Simulator)

```
                        .text
00001000:EF9F0014       LDR R0, =a
00001004:EF9F1014       LDR R1,=b
00001008:EF9F3014       LDR R3, =c
0000100C:E5D14000       LDR R4, [r1]
00001010:E5D05000       LDR R5, [r0]
00001014:E0846005       ADD R6, R4, R5
00001018:E00360B0       STR R6, [r3]

                        .data
00001028:               a: .word 100
00001029:               b: word 200
0000102A:               c: word 0
```



Salient features of ARM Instructions

- Each Instruction is 32 bit wide
- Load-store architecture
- 3-address instructions.
- Conditional execution of every instruction.
- Possible to load/store multiple register at once.
- Possible to combine shift and ALU operations in a single instruction.
- No Memory-Memory Operations

General Format of Instruction

```
MNEMONIC{condition}{S} {Rd},Operand1,Operand2
```

MNEMONIC - Short name of the instruction. ***Eg: ADD,SUB….***

{condition} - Condition that is needed to be met in order for the instruction to be executed ***EG: EQ, MI,GT,LT,LE,AL,NE***

{S} - An optional suffix. If S is specified, the condition flags are updated on the result of the operation . ***Eg: To set N,O, C, V of CPSR***

{Rd} - Register (destination) for storing the result of the instruction

Operand1 - First operand. Either a register or an immediate value

Operand2 - Second (flexible) operand. Can be an immediate value (number) or a register with an optional shift

General Format of Instruction Example

Example 1:

MOV r0, #-10
MOV r1, #-10
ADD r2, r1,r0

```
R0      :-10
R1      :-10
R2      :-20
R3      :0
R4      :0
R5      :0
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):70656
------------------
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
------------------
```

General Format of Instruction Example

Example2:

MOV r0, #-10
MOV r1, #-10
ADDS r2, r1,r0

```
R0      :-10
R1      :-10
R2      :-20
R3      :0
R4      :0
R5      :0
R6      :0
R7      :0
R8      :-10
R9      :0
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):21504
R14(lr):0
R15(pc):70656
------------------
CPSR Register
Negative(N):1
Zero(Z)     :0
Carry(C)    :1
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)    :0
CPU Mode    :System
------------------
```

General Format of Instruction Example

Example 3

```
.text
MOV r0,#-44
MOV r1,#4
 SUBS r2,r0,r1
 ADDMIS r0,r0,r1
.end
```

- Subtract r1 from r0
- Add only if the result of SUB is -ve

(Here result of SUB is –ve so add is executed

```
R0      :-40
R1      :4
R2      :-48
R3      :0
R4      :0
R5      :0
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):70656
------------------
CPSR Register
Negative(N):1
Zero(Z)    :0
Carry(C)   :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :0
CPU Mode   :System
------------------
```

General Format of Instruction Example

Example 4

```
.text
MOV r0,#44
MOV r1,#-4
 SUBS r2,r0,r1
 ADDMIS r0,r0,r1
.end
```

- Subtract r1 from r0
- Add only if the result of SUB is -ve

(Here result of SUB is not –ve so add is not executed

```
R0      :44
R1      :-4
R2      :48
R3      :0
R4      :0
R5      :0
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):70656
------------------
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
------------------
```

General Format of Instruction Example

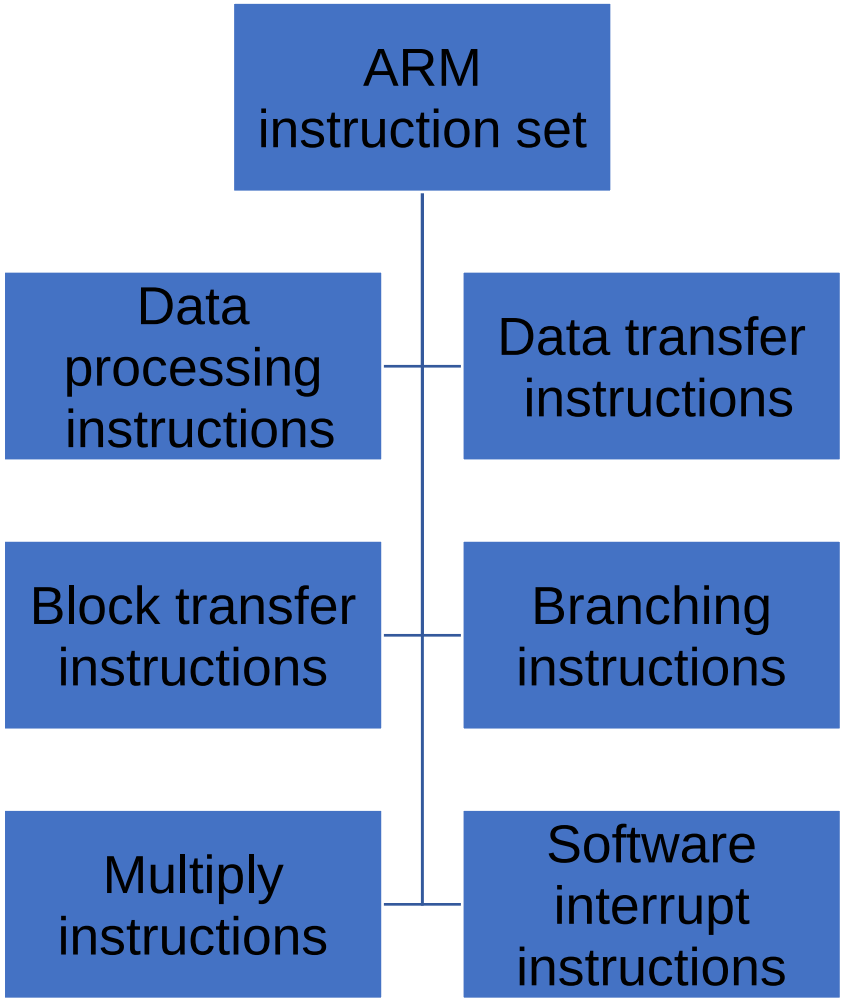
Example 5

```
.text
MOV r0,#44
MOV r1,#4
CMP r1,r0
ADDEQ r0,r0,r1
.end
```

- Compare r1 and r0
- Add only of the comparison is true i.e N=0
  (Here comparison fails i.e N=1)

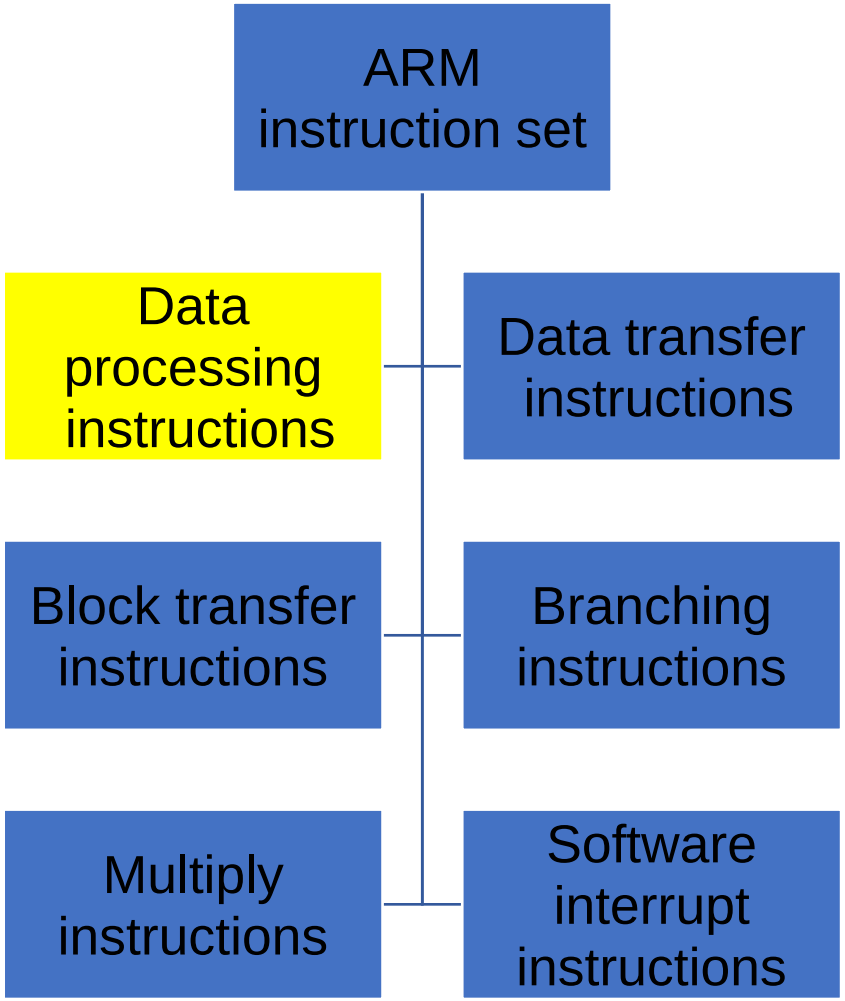
```
R0      :44
R1      :4
R2      :0
R3      :0
R4      :0
R5      :0
R6      :0
R7      :0
R8      :0
R9      :0
R10(sl):0
R11(fp):0
R12(ip):0
R13(sp):21504
R14(lr):0
R15(pc):70656
------------------
CPSR Register
Negative(N):1
Zero(Z)    :0
Carry(C)   :0
Overflow(V):0
IRQ Disable:1
FIQ Disable:1
Thumb(T)   :0
CPU Mode   :System
------------------
```

ARM Instruction Set



Microprocessor & Computer Architecture (µpCA)

Next Session



Click Here for reference

THANK YOU

Dr. D. C. Kiran
Department of Computer Science and Engineering

dckiran@pes.edu

9829935135