Computer Organization And Microprocessors

ENCS2380

By :
Mohammed Saada

Assembly.

- In ARM:
  - The memory is Byte-addressable (e.g. each cell is 1 Byte (8-bits) width.
  - There is 16 Registers:
    - General Purpose Registers ($R_0 - R_{12}$)
    - Special purpose Registers:
      - R_{13} (SP) : Stack pointer.
      - R_{14} (LR) : Link Register.
      - R_{15} (PC) : Program Counter.
  - Registers width 32-bits (word)

- ARM Instructions:

Instruction	Description
ADD Rd, Rn,Op2*	ADD Rn to Op2 and place the result in Rd
ADC Rd, Rn,Op2	ADD Rn to Op2 with Carry and place the result in Rd
AND Rd, Rn,Op2	AND Rn with Op2 and place the result in Rd
BIC Rd, Rn,Op2	AND Rn with NOT of Op2 and place the result in Rd
CMP Rn,Op2	Compare Rn with Op2 and set the status bits of CPSR**
CMN Rn,Op2	Compare Rn with negative of Op2 and set the status bits
EOR Rd, Rn,Op2	Exclusive OR Rn with Op2 and place the result in Rd
MVN Rd,Op2	Place NOT of Op2 in Rd
MOV Rd,Op2	MOVE (Copy) Op2 to Rd
ORR Rd, Rn,Op2	OR Rn with Op2 and place the result in Rd
RSB Rd, Rn,Op2	Subtract Rn from Op2 and place the result in Rd
RSC Rd, Rn,Op2	Subtract Rn from Op2 with carry and place the result in Rd
SBC Rd, Rn,Op2	Subtract Op2 from Rn with carry and place the result in Rd
SUB Rd, Rn,Op2	Subtract Op2 from Rn and place the result in Rd
TEQ Rn,Op2	Exclusive-OR Rn with Op2 and set the status bits of CPSR
TST Rn,Op2	AND Rn with Op2 and set the status bits of CPSR

* Op2 can be an immediate 8-bit value #K which can be 0–255 in decimal, (00–FF in hex). Op2 can also be a register Rm. Rd, Rn and Rm are any of the general purpose registers

** CPSR is discussed later in this chapter

*** The instructions are discussed in detail in the next chapters

Table 2- 1: ALU Instructions Using GPRs

Instruction	Flags Affected
ANDS	C, Z, N
ORRS	C, Z, N
MOVS	C, Z, N
ADDS	C, Z, N, V
SUBS	C, Z, N, V
B	No flags
Note that we cannot put S after B instruction.	

Instruction	Flags Affected
BCS	Branch if C = 1
BCC	Branch if C = 0
BEQ	Branch if Z = 1
BNE	Branch if Z = 0
BMI	Branch if N = 1
BPL	Branch if N = 0
BVS	Branch if V = 1
BVC	Branch if V = 0

Directive	Description
DCB	Allocates one or more bytes of memory, and defines the initial runtime contents of the memory
DCW	Allocates one or more halfwords of memory, aligned on two-byte boundaries, and defines the initial runtime contents of the memory.
DCWU	Allocates one or more halfwords of memory, and defines the initial runtime contents of the memory. The data is not aligned.
DCD	Allocates one or more words of memory, aligned on four-byte boundaries, and defines the initial runtime contents of the memory.
DCDU	Allocates one or more words of memory and defines the initial runtime contents of the memory. The data is not aligned.

ADR directive

To load registers with the addresses of memory locations we can also use the ADR pseudo-instruction which has a better performance

```
ADR Rn,label
```

```
ADR R2, OUR_FIXED_DATA ;point to OUR_FIXED_DATA
```

Instruction (Flags unchanged)		Instruction (Flags updated)	
ADD	Add	**ADDS**	Add and set flags
ADC	Add with carry	**ADCS**	Add with carry and set flags
SUB	SUBS	**SUBS**	Subtract and set flags
SBC	Subtract with carry	**SBCS**	Subtract with carry and set flags
MUL	Multiply	**MULS**	Multiply and set flags
UMULL	Multiply long	**UMULLS**	Multiply Long and set flags
RSB	Reverse subtract	**RSBS**	Reverse subtract and set flags
RSC	Reverse subtract with carry	**RSCS**	Reverse subtract with carry and set flags

Note: The above instruction affect all the Z, C, V and N flag bits of CPSR (current program status register) but the N and V flags are for signed data and are discussed in Chapter 5.

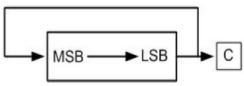
Instruction (Flags Unchanged)	Action	Instruction (Flags Changed)	Hexadecimal
AND	ANDing	**ANDS**	Anding and set flags
ORR	ORRing	**ORS**	Oring and set flags
EOR	Exclusive-ORing	**EORS**	Exclusive Oring and set flags
BIC	Bit Clearing	**BICS**	Bit clearing and set flags

Operation	Destination	Source	Number of shifts
LSR (Shift Right)	Rd	Rn	Immediate value
LSR (Shift Right)	Rd	Rn	register Rm
LSL (Shift Left)	Rd	Rn	Immediate value
LSL (Shift Left)	Rd	Rn	register Rm

Note: Number of shift cannot be more than 32.

Arithmetic shift right ASR

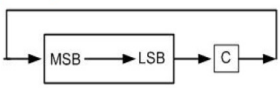
ROR (rotate right)



Rotate left

There is no rotate left option in ARM7 since one can use the rotate right (ROR) to do the job. That means instead of rotating left n bits we can use rotate right 32–n bits to do the job of rotate left

RRX rotate right through carry



Operation	Destination	Source	Number of Rotates
ROR (Rotate Right)	Rd	Rn	Immediate value
ROR (Rotate Right)	Rd	Rn	register Rm
RRX (Rotate Right Through Carry)	Rd	Rn	1 bit

ASCII to packed BCD conversion

```
MOV R1,#0x37     ;R1 = 0x37
MOV R2,#0x32     ;R2 = 0x32
AND R1,R1,#0x0F ;mask 3 to get unpacked BCD
AND R2,R2,#0x0F ;mask 3 to get unpacked BCD
MOV R3,R2,LSL #4 ;shift R2 4 bits to left to get R3 = 0x20
ORR R4,R3,R1       ;OR them to get packed BCD, R4 = 0x27
```

Packed BCD to ASCII conversion

```
MOV R0,#0x29
AND R1,R0,#0x0F      ;mask upper four bits
ORR R1,R1,#0x30      ;combine with 30 to get ASCII
MOV R2,R0,LSR #04 ;shift right 4 bits to get unpacked BCD
ORR R2,R2,#0x30      ;combine with 30 to get ASCII
```

⇒ Branches Instructions :

Instruction		Action
BCS/BHS	branch if carry set/branch if higher or same	Branch if C = 1
BCC/BLO	branch if carry clear/branch lower	Branch if C = 0
BEQ	branch if equal	Branch if Z = 1
BNE	branch if not equal	Branch if Z = 0
BLS	branch if less or same	Branch if Z = 1 or C = 0
BHI	branch if higher	Branch if Z = 0 and C = 1

unsigned

BL (Branch and link)

When a subroutine is called using BL instruction, first the processor saves the address of the instruction just below the BL instruction on the R14 register (LR, linker register), and then control is transferred to that subroutine

	Instruction	Action
BEQ	branch equal	Branch if Z = 1
BNE	Branch not equal	Branch if Z = 0
BMI	Branch minus (branch negative)	Branch if N = 1
BPL	Branch plus (branch positive)	Branch if N = 0
BVS	Branch if V set (branch overflow)	Branch if V = 1
BVC	Branch if V clear (branch if no overflow)	Branch if V = 0
BGE	Branch greater than or equal	Branch if N = V
BLT	Branch less than	Branch if N ≠ V
BGT	Branch greater than	Branch if Z = 0 and N = V
BLE	Branch less than or equal	Branch if Z = 1 or N ≠ V

signed

Signed number comparison

CMP Rn, Op2

Op2 > Rn V = N
Op2 = Rn Z = 1
Op2 < Rn N ≠ V

Summary of ARM's Indexed Addessing Modes

Addressing Mode	Assembly Mnemonic	Effective address	FinalValue in R1
Pre-indexed, base unchanged	LDR R0, [R1, #d]	R1 + d	R1
Pre-indexed, base updated	LDR R0, [R1, #d]!	R1 + d	R1 + d
Post-indexed, base updated	LDR R0, [R1], #d	R1	R1 + d