Lab 4

Dr. Upama Kabir
Professor

Computer Science and Engineering, University of Dhaka

Feb 20, 2023

Outline

1. Data Processing Instruction Set

2. Tasks to Complete

Arithmetic Data Operation

Table 1: Arithmetic Data Operation

Operation	**Mnemonic**	**Index Mode**
Add	ADD	ADD R_d, R_n, R_m
Add	ADD	ADD $R_d, R_n, \#immed$
Add with Carry	ADC	ADC R_d, R_n, R_m
Add with Carry	ADC	ADC $R_d, \#immed$
Subtract	SUB	SUB R_d, R_m, R_n
Subtract	SUB	SUB $R_d, R_n, \#immed$
Subtract with Borrow	SBC	SBC $R_d, R_n, \#immed$
Subtract with Borrow	SBC	SBC R_d, R_n, R_m
Reverse Subtract	RSB	RSB R_d, R_n, R_m

Arithmetic Data Operation

Table 2: Arithmetic Data Operation

Operation	**Mnemonic**	**Index Mode**
Multiply	MUL	MUL R_d, R_n, R_m
Unsigned Division	UDIV	UDIV $R_d, R_n, \#immed$
Signed Division	ADC	SDIV R_d, R_n, R_m
Multiply Accumulate	MLA	MLA R_d, R_n, R_m, R_as
Multiply Subtract	MLS	MLS R_d, R_n, R_m, R_as
S Signed Multiply Long	SMULL	SMULL $Rd_{lo}, Rd_{hi}, R_d, R_n$
Unsigned Multiply Long	UMULL	UMULL $Rd_{lo}, Rd_{hi}, R_d, R_n$

Logic Operation

Table 3: Logic Operation

Operation	Mnemonic	Index Mode
Bitwise AND	AND	AND R_d, R_n
Bitwise AND	AND	AND R_d, R_n, R_m
Bitwise OR	ORR	ORR R_d, R_n
Bitwise Bit Clear	BIC	BIC R_d, R_n, R_m
Bitwise OR NOT Clear	ORN	ORN R_d, R_n, R_m
Bitwise Exclusive OR	EOR	EOR $R_d, R_n, R_m, R_a s$
S Bitwise NOT	MVN	MVN R_d

Shift Operation

Table 4: Shift Operation

Operation	Mnemonic	Index Mode
Arithmetic Shift Right	ASR	ASR R_d, R_n
Logical Shift Left	LSL	LSL R_d, R_n
Logical Shift Right	LSR	LSR R_d, R_n
Rotate Right	ROR	ROR R_d, R_n, R_m
Reverse Bit	RBIT	RBIT R_d, R_n

Task to Complete

- Write an assembly language to perform all the logical operations (AND,OR,NOR,NAND,XOR,XNOR) on two 16-bit variables. Repeat it for two 32-bit variables.
- Write an assembly language to perform all the shift operations (LSR, ASR, LSL) on a 32-bit variable.
- Write an assembly language to perform all the arithmetic operations (Addition, Subtraction, Division and Multiplication) on two variables. Restrict input values to avoid overflow. Repeat the same operations to handle overflow.
- Write an assembly language program to find the average of n numbers.
- Write an assembly language program to find the largest among n different numbers.
- Write an assembly language program to find the average of n numbers using function call.