

Data Transfer Instruction- Keil Simulation

Pre-indexing

```
1  AREA PRE_INDEX, CODE, READONLY
2  ENTRY
3      ADR R1, TABLE
4      LDR R2, [R1, #4]
5  LP B LP
6
7
8  TABLE DCD 1, 2, 3, 4, 5
9      END
```

Current	
R0	0x00000000
R1	0x0000000C
R2	0x00000002
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000008
CPSR	0x000000D3
SPSR	0x00000000

Disassembly

```
3:      ADR R1, TABLE
0x00000000  E28F1004  ADD      R1, PC, #0x00000004
4:      LDR R2, [R1, #4]
0x00000004  E5912004  LDR      R2, [R1, #0x0004]
5:  LP B LP
→ 0x00000008  EAfffffe  B        0x00000008
```

Memory 1

Address: 0X00

```
0x00000000: 04 10 8F E2
0x00000004: 04 20 91 E5
0x00000008: FE FF FF EA
0x0000000C: 01 00 00 00
0x00000010: 02 00 00 00
0x00000014: 03 00 00 00
0x00000018: 04 00 00 00
0x0000001C: 05 00 00 00
0x00000020: 00 00 00 00
0x00000024: 00 00 00 00
0x00000028: 00 00 00 00
```

Call Stack + Locals

Memory 1

Auto-indexing

```
1      AREA AUTO_INDEX, CODE, READONLY
2 ENTRY
3      ADR R1, TABLE
4      LDR R2, [R1, #4]!
5      STR R2, [R1, #12]!
6 LP B LP
7
8
9 TABLE DCD 1, 2, 3, 4, 5
10     END
```

Registers

Register	Value
Current	
R0	0x00000000
R1	0x00000020
R2	0x00000002
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x0000000C
CPSR	0x000000D3

Disassembly

0x00000000	E28F1008	ADD	R1, PC, #0x00000008
4:		LDR R2, [R1, #4]!	
0x00000004	E5B12004	LDR	R2, [R1, #0x0004]!
5:		STR R2, [R1, #12]!	
0x00000008	E5A1200C	STR	R2, [R1, #0x000C]!
6: LP B LP			
→ 0x0000000C	EAFFFFFE	B	0x0000000C
0x00000010	00000001	ANDEQ	R0, R0, R1
0x00000014	00000002	ANDEQ	R0, R0, R2
0x00000018	00000003	ANDEQ	R0, R0, R3
0x0000001C	00000004	ANDEQ	R0, R0, R4
0x00000020	00000005	ANDEQ	R0, R0, R5
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

Memory 1

Address: 0X00

0x00000000:	08	10	8F	E2
0x00000004:	04	20	B1	E5
0x00000008:	0C	20	A1	E5
0x0000000C:	FE	FF	FF	EA
0x00000010:	01	00	00	00
0x00000014:	02	00	00	00
0x00000018:	03	00	00	00
0x0000001C:	04	00	00	00
0x00000020:	05	00	00	00
0x00000024:	00	00	00	00
0x00000028:	00	00	00	00
0x0000002C:	00	00	00	00
0x00000030:	00	00	00	00
0x00000034:	00	00	00	00
0x00000038:	00	00	00	00

Post-indexing

```
1 AREA POST_INDEX, CODE, READONLY
2 ENTRY
3     ADR R1, TABLE
4     LDR R2, [R1], #4
5 LP B LP
6
7
8 TABLE DCD 1, 2, 3, 4, 5
9     END
```

The screenshot displays an ARM development tool interface with three main panels: Registers, Disassembly, and Memory 1.

Registers Panel: A table showing the current state of 16 registers (R0-R15) and CPSR/SPSR. R15 (PC) is highlighted as the current register.

Register	Value
R0	0x00000000
R1	0x00000010
R2	0x00000001
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000008
CPSR	0x000000D3
SPSR	0x00000000

Disassembly Panel: Shows the assembly code being executed. The instruction at address 0x00000008 is highlighted: `B 0x00000008`.

Address	Disassembly
0x00000000	3: ADR R1, TABLE
0x00000004	4: LDR R2, [R1], #4
0x00000008	5: LP B LP
0x0000000C	00000001 ANDEQ R0, R0, R1
0x00000010	00000002 ANDEQ R0, R0, R2
0x00000014	00000003 ANDEQ R0, R0, R3
0x00000018	00000004 ANDEQ R0, R0, R4
0x0000001C	00000005 ANDEQ R0, R0, R5
0x00000020	00000000 ANDEQ R0, R0, R0
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

Memory 1 Panel: Shows the memory contents starting from address 0x00000000. The value at address 0x0000000C is highlighted in red.

Address	Value (Hex)
0x00000000	04 10 8F E2
0x00000004	04 20 91 E4
0x00000008	FE FF FF EA
0x0000000C	01 00 00 00
0x00000010	02 00 00 00
0x00000014	03 00 00 00
0x00000018	04 00 00 00
0x0000001C	05 00 00 00
0x00000020	00 00 00 00
0x00000024	00 00 00 00
0x00000028	00 00 00 00
0x0000002C	00 00 00 00
0x00000030	00 00 00 00
0x00000034	00 00 00 00
0x00000038	00 00 00 00
0x0000003C	00 00 00 00

The bottom status bar shows the current file is `pre_index.asm`.

Multiple Register Transfer – Load Instruction

```
1 AREA MGT_LD, CODE, READONLY
2 ENTRY
3 ADR R1, TABLE
4 LDMIA R1, {R2-R6}
5 LP B LP
6
7
8 TABLE DCD 1, 2, 3, 4, 5
9 END
```

The screenshot displays a debugger interface with three main panels: Registers, Disassembly, and Memory 1.

Registers Panel: Shows the current state of registers R0 through R15 (PC) and CPSR. R15 (PC) is highlighted with a value of 0x00000008.

Disassembly Panel: Shows the assembly code being executed. The instruction at address 0x00000008 is highlighted: `5: LP B LP`. The instruction at address 0x00000004 is `4: LDMIA R1, {R2-R6}`.

Memory 1 Panel: Shows the memory contents starting from address 0x00000000. The memory is organized into a table with columns for address, hex value, and decimal value.

Address	Hex Value	Decimal Value
0x00000000	04 10 8F E2	
0x00000004	7C 00 91 E8	
0x00000008	FE FF FF EA	
0x0000000C	01 00 00 00	1
0x00000010	02 00 00 00	2
0x00000014	03 00 00 00	3
0x00000018	04 00 00 00	4
0x0000001C	05 00 00 00	5
0x00000020	00 00 00 00	
0x00000024	00 00 00 00	
0x00000028	00 00 00 00	
0x0000002C	00 00 00 00	
0x00000030	00 00 00 00	
0x00000034	00 00 00 00	
0x00000038	00 00 00 00	
0x0000003C	00 00 00 00	

Multiple Register Transfer – Store Instruction

```
1 AREA MGT_STR, CODE, READONLY
2 ENTRY
3 LDR R1,=0X10
4 LDR R2,=0X20
5 LDR R3,=0X30
6 LDR R4,=0X40
7 LDR R5,=0X50
8 ADR R0, TABLE
9 STMIA R0!, {R1-R5}
10 LP B LP
11
12
13 TABLE DCD 1, 2, 3, 4, 5
14 END
```

The screenshot displays a debugger interface with three main panels: Registers, Disassembly, and Memory 1.

Registers Panel: Shows the current state of 16 registers (R0-R15) and CPSR/SPSR. R0 is 0x00000034, R1 is 0x00000010, R2 is 0x00000020, R3 is 0x00000030, R4 is 0x00000040, R5 is 0x00000050, and R15 (PC) is 0x0000001C.

Disassembly Panel: Shows the assembly code being executed. The current instruction is at address 0x0000001C: `EAFFFFFFE B 0x0000001C`, which is a branch instruction `B` to the label `LP`. The previous instruction was `STMIA R0!, {R1-R5}` at address 0x00000018.

Memory 1 Panel: Shows the memory contents starting from address 0x00. The memory contains a sequence of values: 10 10 A0 E3, 20 20 A0 E3, 30 30 A0 E3, 40 40 A0 E3, 50 50 A0 E3, 04 00 8F E2, 3E 00 A0 E8, FE FF FF EA, 01 00 00 00, 02 00 00 00, 03 00 00 00, 04 00 00 00, 05 00 00 00, 00 00 00 00, 00 00 00 00, 00 00 00 00.

Thank you