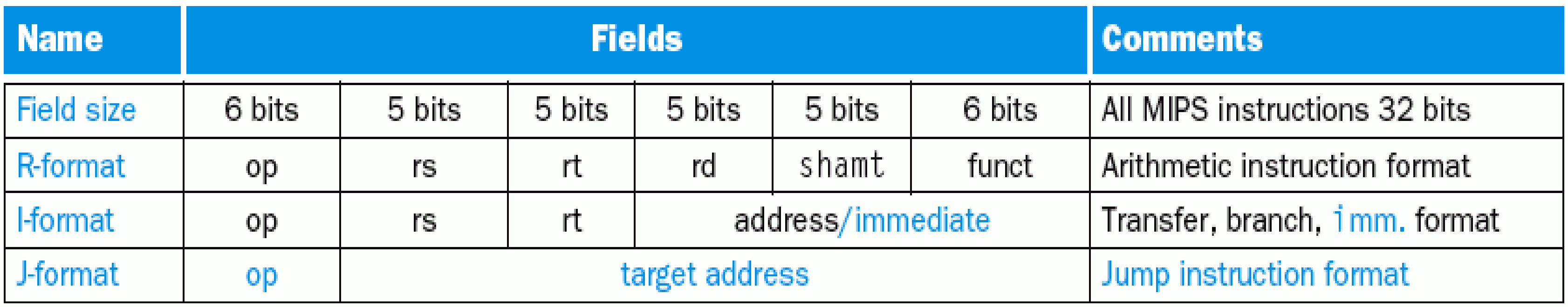
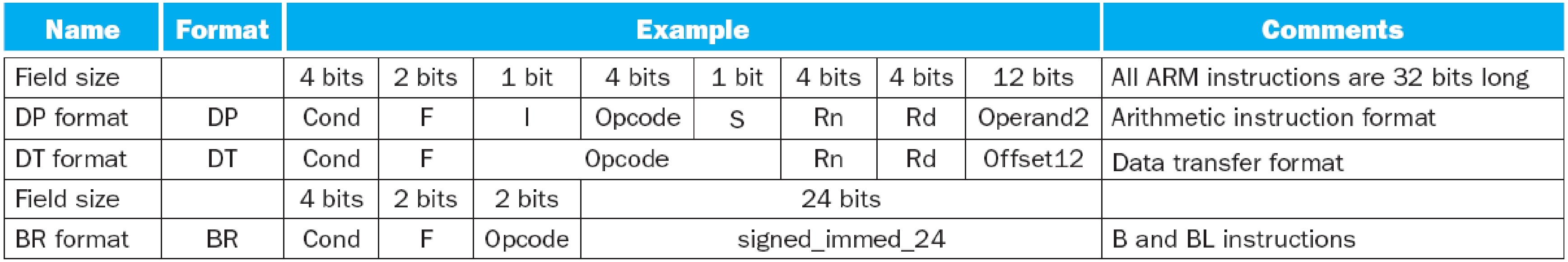
INSTRUCTION FORMAT

MIPS instruction format



Name	Fields						Comments
Field size	6 bits	5 bits	5 bits	5 bits	5 bits	6 bits	All MIPS instructions 32 bits
R-format	op	rs	rt	rd	shamt	funct	Arithmetic instruction format
I-format	op	rs	rt	address/immediate			Transfer, branch, imm. format
J-format	op	target address					Jump instruction format

ARM instruction format



Name	Format	Example								Comments
Field size		4 bits	2 bits	1 bit	4 bits	1 bit	4 bits	4 bits	12 bits	All ARM instructions are 32 bits long
DP format	DP	Cond	F	I	Opcode	S	Rn	Rd	Operand2	Arithmetic instruction format
DT format	DT	Cond	F	Opcode			Rn	Rd	Offset12	Data transfer format
Field size		4 bits	2 bits	2 bits	24 bits					
BR format	BR	Cond	F	Opcode	signed_immed_24					B and BL instructions

Opcode : Basic operation of the instruction, traditionally called the opcode

Rd : The register destination operand. It gets the result of the operation

Rn : The first register source operand

Operand2: The second source operand

I : Immediate. If 0, the second source operand is a register. If 1, the second source operand is a 12-bit immediate

S : Set Condition Code.

Cond : Condition

F : Instruction Format. This field allows ARM to differentiate instruction formats when needed

DP Instruction Example

Cond	F	I	Opcode	S	Rn	Rd	Operand2
4 bits	2 bits	1 bits	4 bits	1 bits	4 bits	4 bits	12 bits

```
ADD r5,r1,r2 ; r5 = r1 + r2
```

14	0	0	4	0	1	5	2
4 bits	2 bits	1 bits	4 bits	1 bits	4 bits	4 bits	12 bits

$$11100000100000010101000000000010_2$$

ARM Data Transfer (DT) Instruction

Cond	F	Opcode	Rn	Rd	Offset12
4 bits	2 bits	6 bits	4 bits	4 bits	12 bits

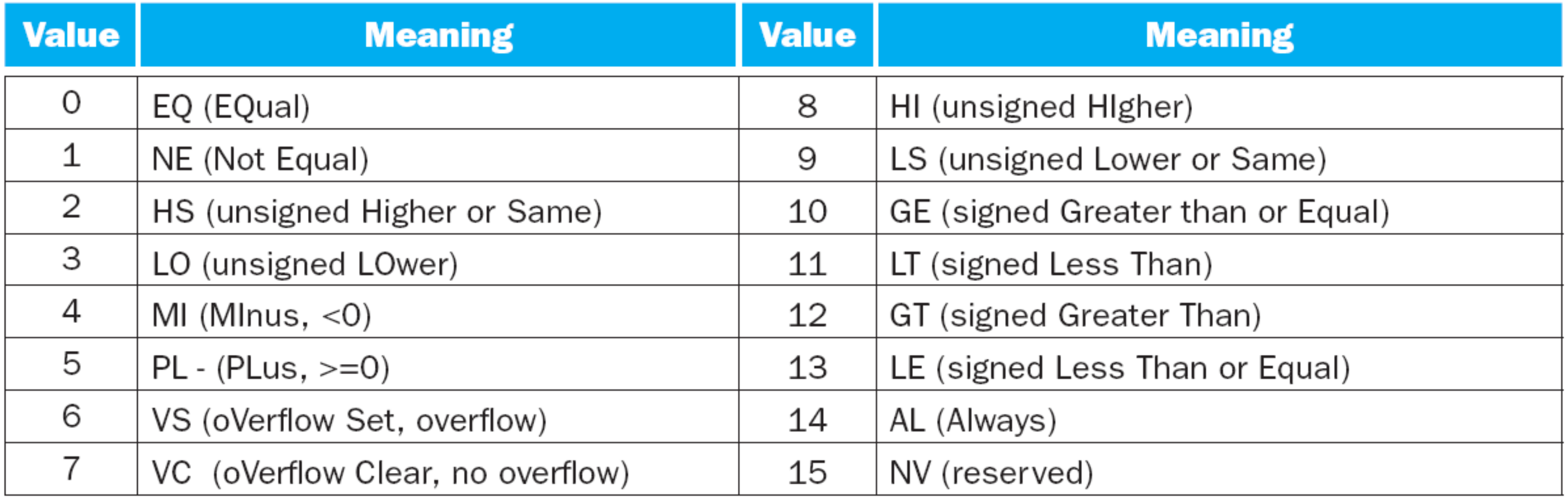
LDR r5, [r3, #32] ; Temporary reg r5 gets A[8]

14	1	24	3	5	32
4 bits	2 bits	6 bits	4 bits	4 bits	12 bits

Branch Instruction format

Condition	12	address
4 bits	4 bits	24 bits

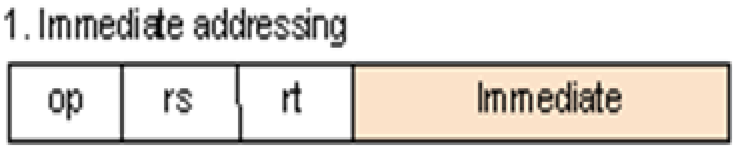
- Encoding of options for Condition field



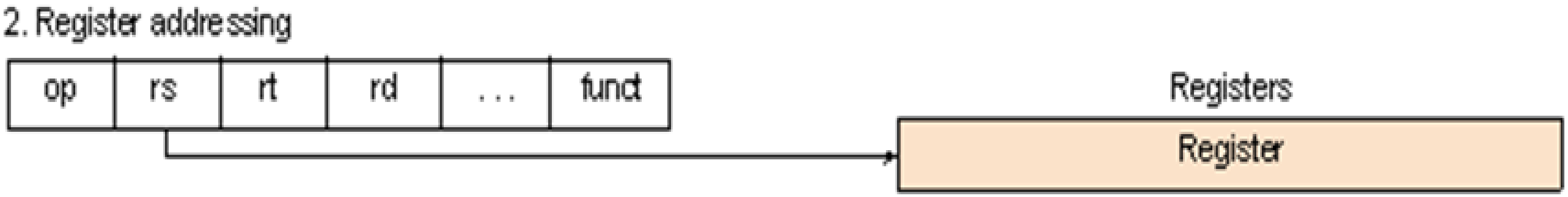
Value	Meaning	Value	Meaning
0	EQ (EQual)	8	HI (unsigned HIgher)
1	NE (Not Equal)	9	LS (unsigned Lower or Same)
2	HS (unsigned Higher or Same)	10	GE (signed Greater than or Equal)
3	LO (unsigned LOwer)	11	LT (signed Less Than)
4	MI (MInus, <0)	12	GT (signed Greater Than)
5	PL - (PLus, >=0)	13	LE (signed Less Than or Equal)
6	VS (oVerflow Set, overflow)	14	AL (Always)
7	VC (oVerflow Clear, no overflow)	15	NV (reserved)

ADDRESSING MODE

MIPS: Addressing Modes (1-2)

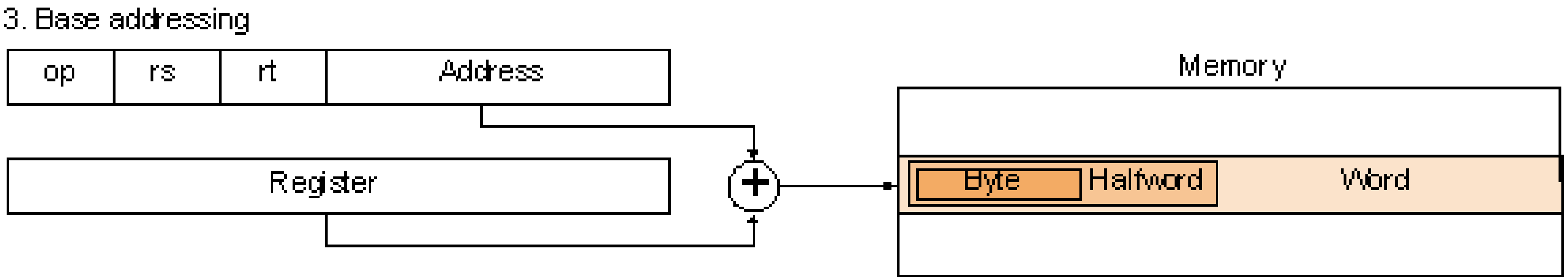


- Operand is a constant from the instruction itself



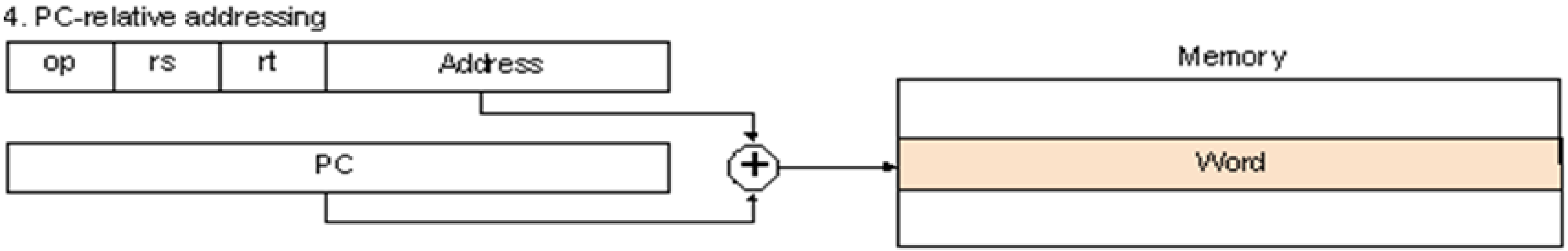
- Data is stored in a register and the register number is given by the instruction

MIPS: Addressing Modes (3)

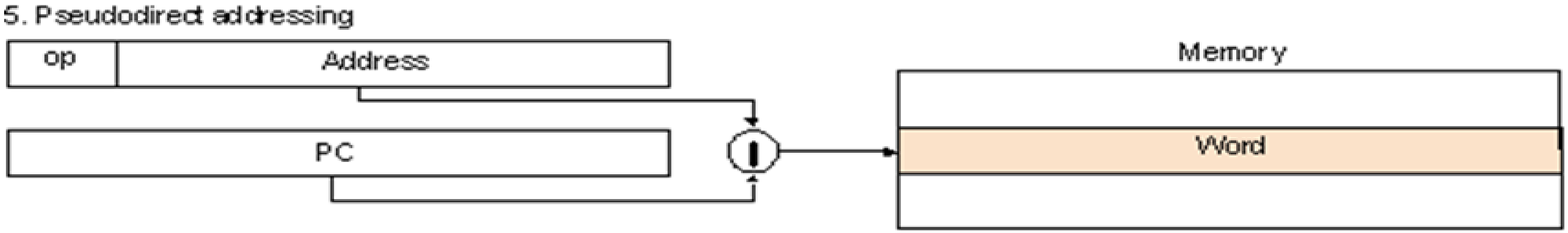


- A special register – base register – is used to hold a base address
- This is also sometimes called the displacement addressing

MIPS: Addressing Modes (4-5)



- The address is the sum of the PC and a constant in the instruction



- 26 bits of the instruction is concatenated with upper PC

ARM: Addressing Mode (1-3 of 12)

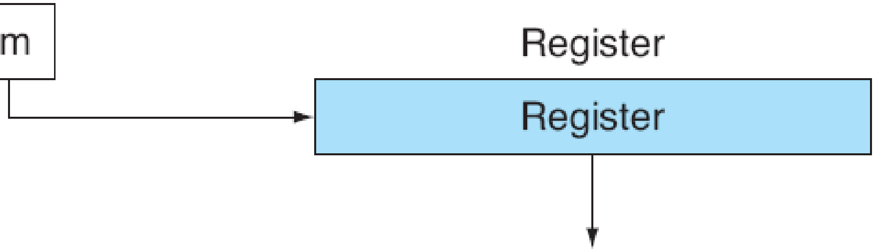
1. Immediate: ADD r2, r0, #5

cond	f	opcode	rn	rd	Immediate

operand is a constant

2. Register: ADD r2, r0, r1

cond	f	opcode	rn	rd	. . .	rm

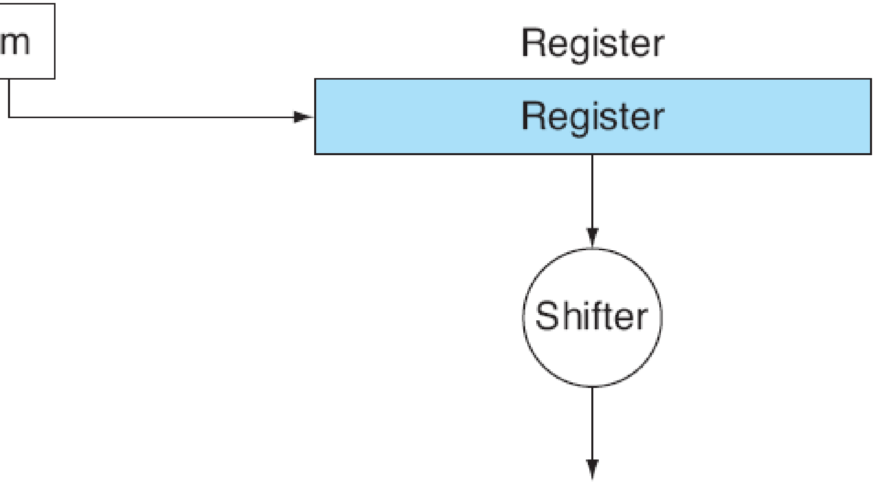

Register

Register

operand is a register

3. Scaled register: ADD r2, r0, r1, LSL #2

cond	f	opcode	rn	rd	. . .	rm


Register

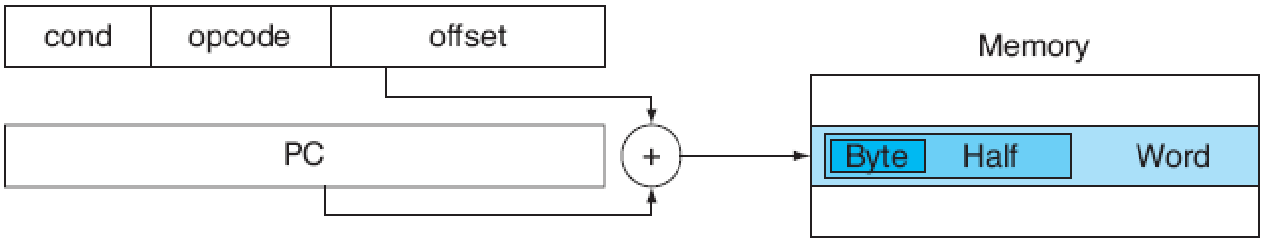
Register

Shifter

register is shifted first

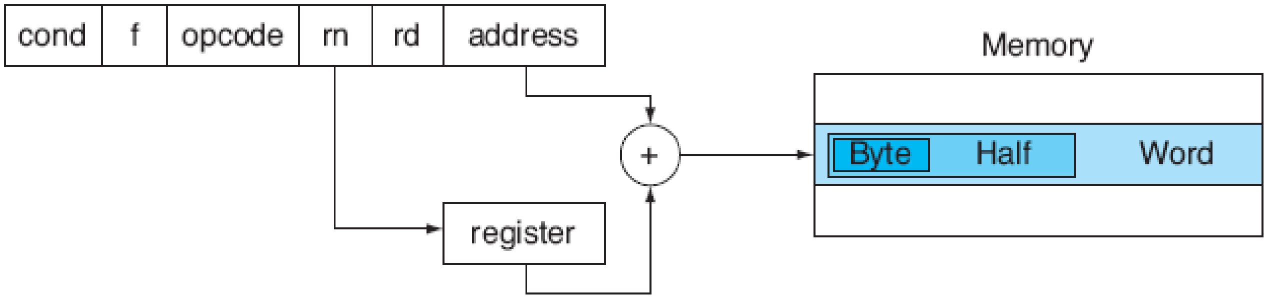
ARM: Addressing Mode (4-6 of 12)

4. PC-relative: BEQ 1000



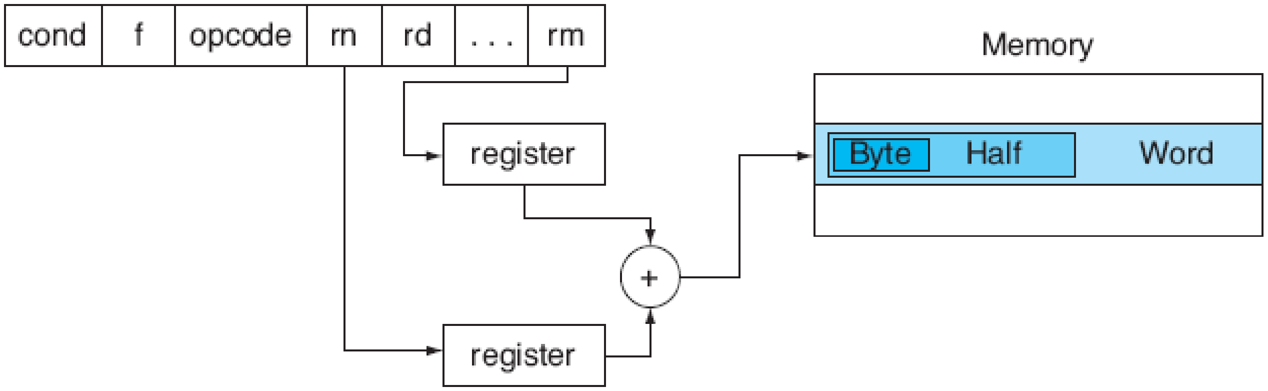
branch address is the sum of the PC and a constant

5. Immediate offset: LDR r2, [r0, #8]



constant address is added to the base reg.

6. Register offset: LDR r2, [r0, r1]

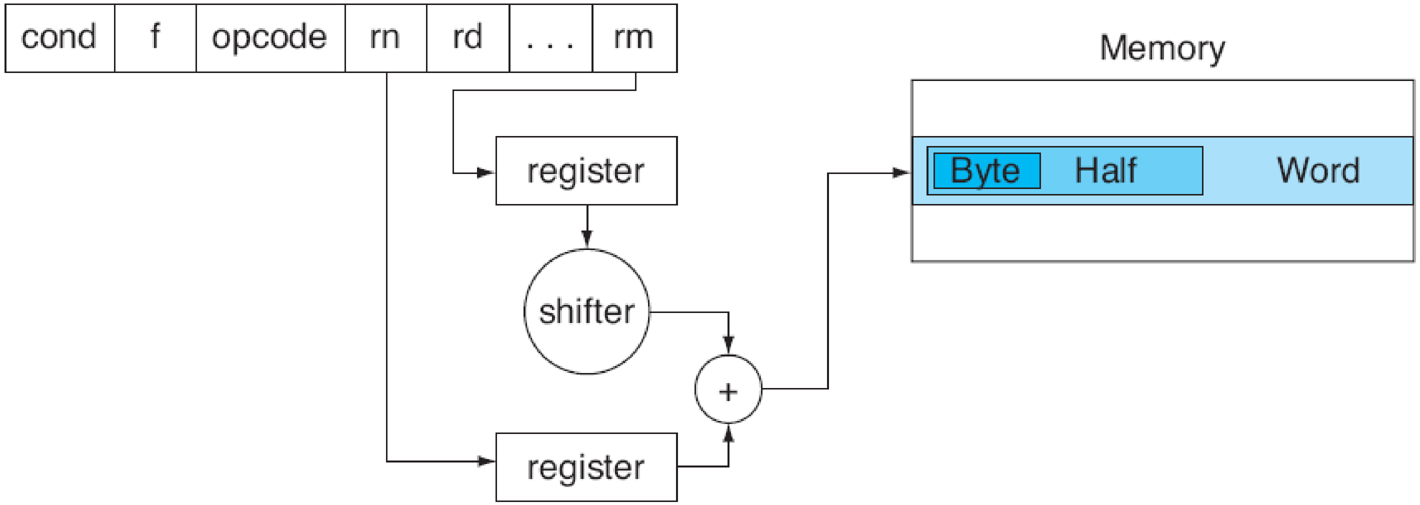


another reg is added to the base reg

ARM: Addressing Mode (7-8 of 12)

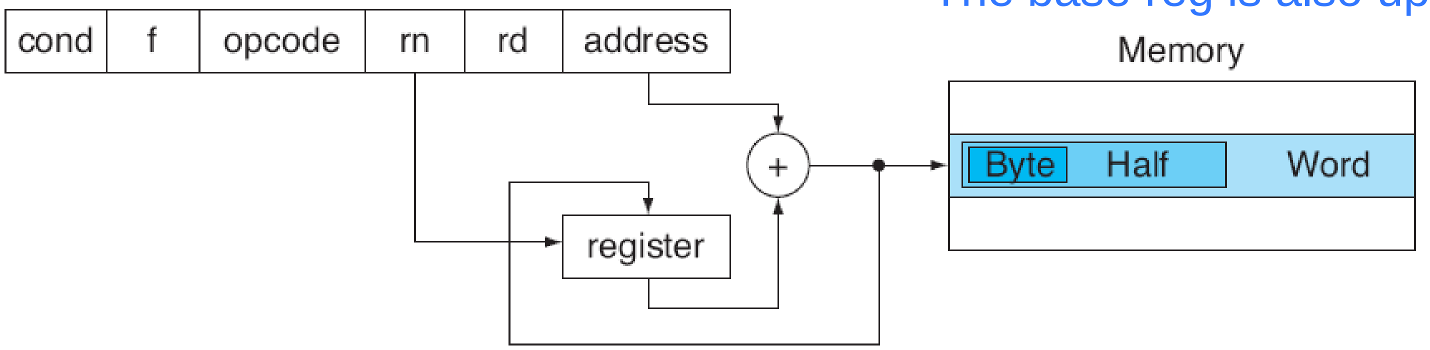
7. Scaled register offset: LDR r2, [r0, r1, LSL #2]

Register offset with scaling



8. Immediate offset pre-indexed: LDR r2, [r0, #4]!

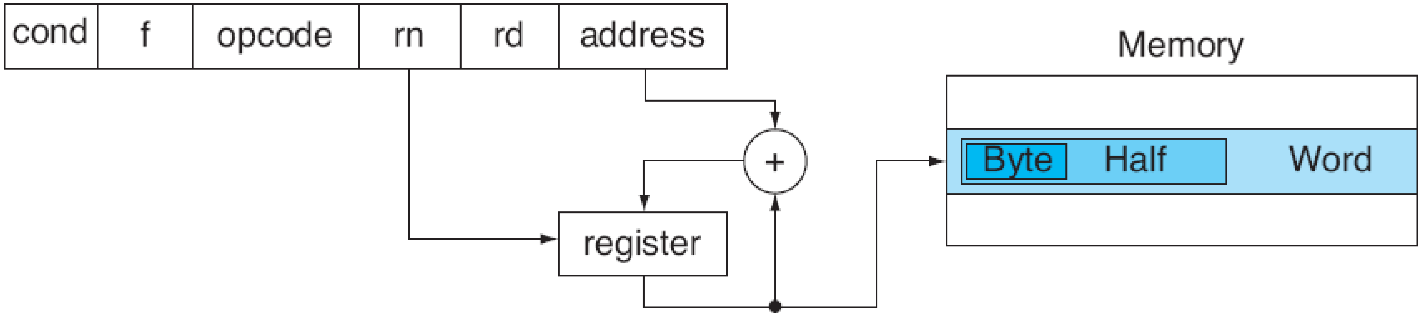
The base reg is also updated



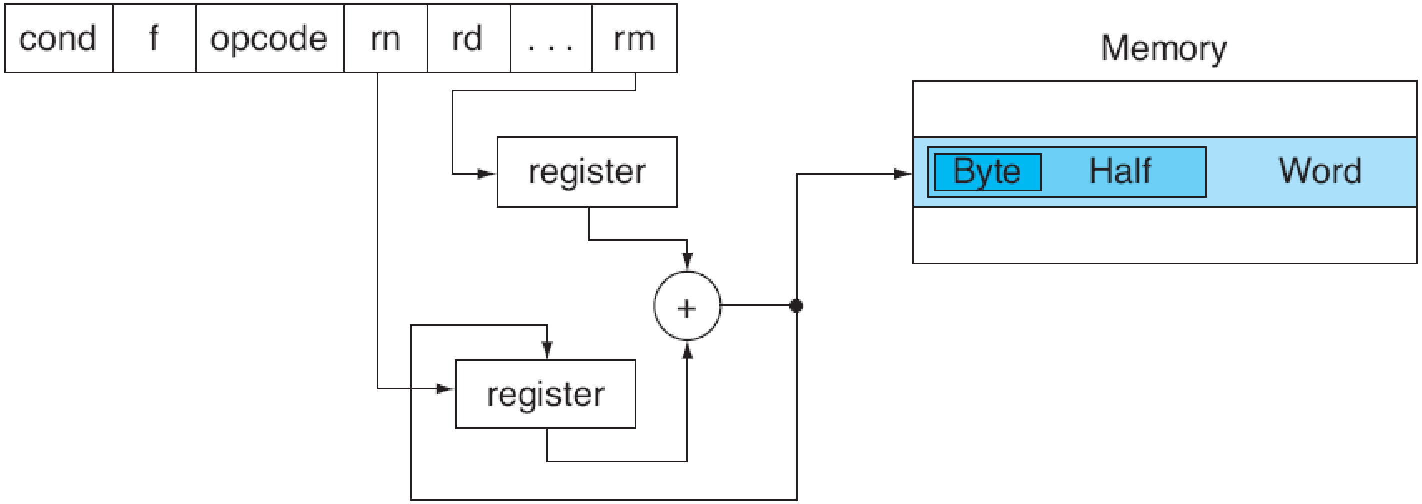
ARM: Addressing Mode (9-10 of 12)

The base reg is also updated

9. Immediate offset post-indexed: LDR r2, [r0], #4

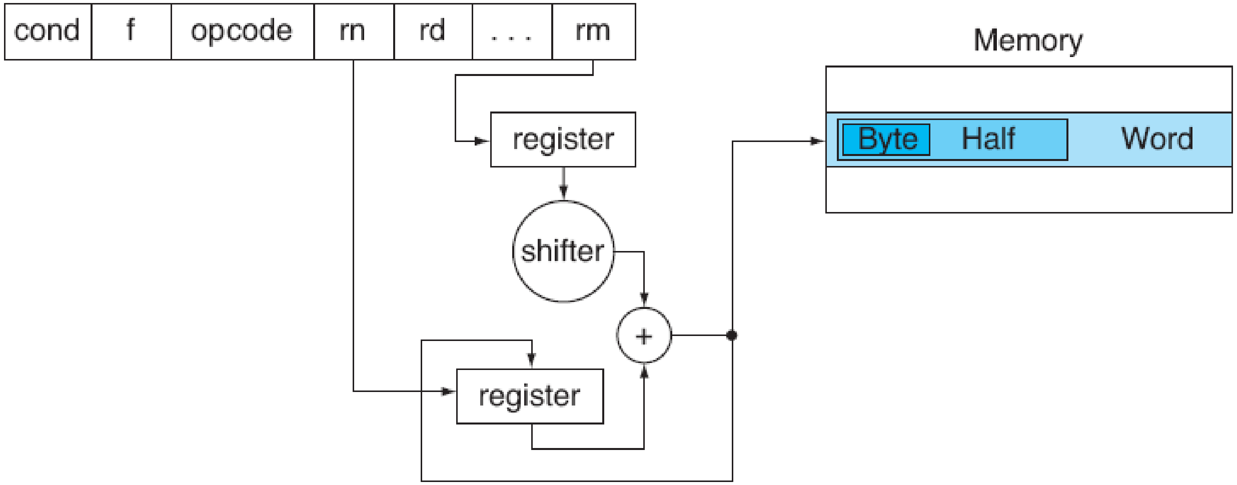


10. Register offset pre-indexed: LDR r2, [r0, r1]!

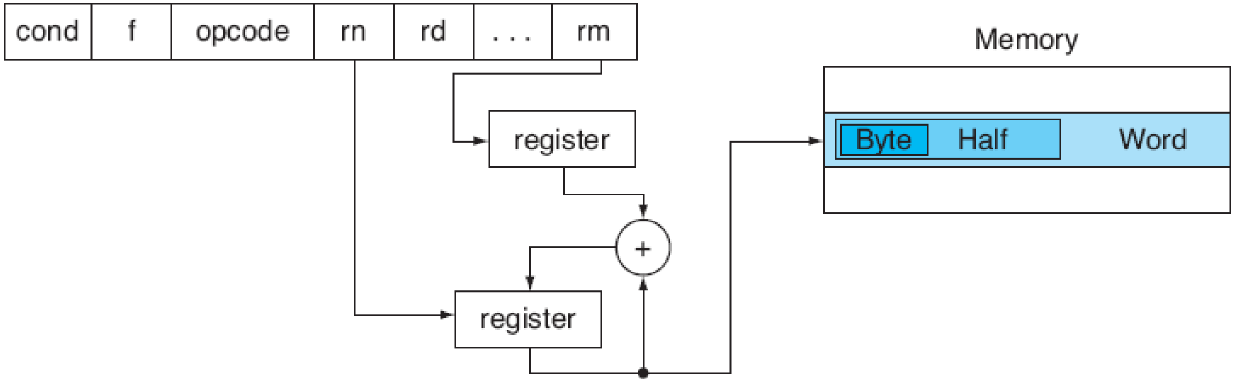


ARM: Addressing Mode (11-12)

11. Scaled register offset pre-indexed: LDR r2, [r0, r1, LSL #2]!



12. Register offset post-indexed: LDR r2, [r0], r1



ARM vs. MIPS

- ARM: the most popular embedded core
- Similar basic set of instructions to MIPS

	ARM	MIPS
Date announced	1985	1985
Instruction size	32 bits	32 bits
Address space	32-bit flat	32-bit flat
Data alignment	Aligned	Aligned
Data addressing modes	9	3
Registers	15 × 32-bit	31 × 32-bit
Input/output	Memory mapped	Memory mapped