

ADDRESSING MODES*

Pedro Dinis
António Espírito Santo
Bruno Ribeiro

This work is produced by The Connexions Project and licensed under the
Creative Commons Attribution License [†]

Addressing modes

The MSP430 supports seven addressing modes for the source operand and four addressing modes for the destination operand (see below). The following sections describe each of the addressing modes, with a brief description, an example and the number of CPU clock cycles required for an instruction, depending on the instruction format and the addressing modes used.

Mode	Source operand	Destination operand	Description
Register mode	X	X	Single cycle
Indexed mode	X	X	Table processing
Symbolic mode	X	X	Easy to read code, PC relative
Absolute mode	X	X	Directly access any memory location
Indirect register mode	X		Access memory with pointers
Indirect auto increment mode	X		Table processing
Immediate mode	X		Unrestricted constant values

Table 1

Before describing the addressing modes, it is important to mention the clock cycles required by interrupts and resets.

*Version 1.1: May 19, 2009 12:26 pm -0500

[†]<http://creativecommons.org/licenses/by/3.0/>

Action	Cycles	Length (words)
Return from interrupt	5	1
Interrupt accepted	6	-
Watchdog timer reset	4	-
Hard reset	4	-

Table 2

1 Register Mode

Register mode operations work directly on the processor registers, R4 through R15, or on special function registers, such as the program counter or status register. They are very efficient in terms of both instruction speed and code space.

Description: Register contents are operands.

Source mode bits: As = 00 (source register defined in the opcode).

Destination mode bit: Ad=0 (destination register defined in the opcode).

Syntax: Rn.

Length: One or two words.

Comment: Valid for source and destination.

Example 1: Move (copy) the contents of source (register R4) to destination (register R5). Register R4 is not affected.

Before operation: R4=A002h R5=F50Ah PC = PC_{pos}

Operation: MOV R4, R5

After operation: R4=A002h R5=A002h PC = PC_{pos} + 2

The first operand is in register mode and depending on the second operand mode, the cycles required to complete an instruction will differ. The next table shows the cycles required to complete an instruction, depending on the second operand mode.

Operands	2 nd operand mode	Operator	Cycles	Length (words)
2	Register	Any	1*	1
2	Indexed, Symbolic or Absolute	Any	4	2
1	N/A	RRA, RRC, SWPB or SXT	1	1
1	N/A	PUSH	3	1
1	N/A	CALL	4	1

Table 3

2 Indexed mode

The Indexed mode commands are formatted as X(Rn), where X is a constant and Rn is one of the CPU registers. The absolute memory location X+Rn is addressed. Indexed mode addressing is useful for applications such as lookup tables.

Description: (Rn + X) points to the operand. X is stored in the next word.

Source mode bits: As = 01 (memory location is defined by the word immediately following the opcode).

Destination mode bit: Ad=1 (memory location is defined by the word immediately following the opcode).

Syntax: X(Rn).

Length: Two or three words.

Comment: Valid for source and destination.

Example 2: Move (copy) the contents at source address (F000h + R5) to destination (register R4).

Before operation: R4=A002h R5=050Ah Loc:0xF50A=0123h

Operation: MOV F000h(R5), R4

After operation: R4=0123h R5=050Ah Loc:0xF50A=0123h

Operands	2 nd operand mode	Operator	Cycles	Length (words)
2	Register	Any	3	2
2	Indexed, Symbolic or Absolute	Any	6	3
1	N/A	RRA, RRC, SWPB or SXT	4	2
1	N/A	CALL or PUSH	5	2

Table 4

3 Symbolic mode

Symbolic mode allows the assignment of labels to fixed memory locations, so that those locations can be addressed. This is useful for the development of embedded programs.

Description: (PC + X) points to the operand. X is stored in the next word. Indexed mode X(PC) is used.

Source mode bits: As = 01 (memory location is defined by the word immediately following the opcode).

Destination mode bit: Ad=1 (memory location is defined by the word immediately following the opcode).

Syntax: ADDR.

Length: Two or three words.

Comment: Valid for source and destination.

Example 3: Move the content of source address XPT (x pointer) to the destination address YPT (y pointer).

Before operation: XPT=A002h Location YPT=050Ah

Operation: MOV XPT, YPT

After operation: XPT= A002h Location YPT=A002h

Operands	2 nd operand mode	Operator	Cycles	Length (words)
2	Register	Any	3	2
2	Indexed, Symbolic or Absolute	Any	6	3
1	N/A	RRA, RRC, SWPB or SXT	4	2
1	N/A	CALL or PUSH	5	2

Table 5

4 Absolute mode

Similar to Symbolic mode, with the difference that the label is preceded by “&”.

Description: The word following the instruction contains the absolute address. X is stored in the next word. Indexed mode X(SR) is used.

Source mode bits: As = 01 (memory location is defined by the word immediately following the opcode).

Destination mode bit: Ad=1 (memory location is defined by the word immediately following the opcode).

Syntax: &ADDR.

Length: Two or three words.

Comment: Valid for source and destination.

Example 4: Move the content of source address XPT to the destination address YPT.

Before operation: Location XPT=A002h Location YPT=050Ah

Operation: MOV &XPT, &YPT

After operation: Location XPT= A002h Location YPT=A002h

Operands	2 nd operand mode	Operator	Cycles	Length (words)
2	Register	Any	3	2
2	Indexed, Symbolic or Absolute	Any	6	3
1	N/A	RRA, RRC, SWPB or SXT	4	2
1	N/A	CALL or PUSH	5	2

Table 6

5 Indirect register mode

The data word addressed is located in the memory location pointed to by Rn. Indirect mode is not valid for destination operands, but can be emulated with the indexed mode format 0(Rn).

Description: Rn is used as a pointer to the operand.

Source mode bits: As = 10.

Syntax: @Rn.

Length: One or two words.

Comment: Valid only for source operand. The substitute for destination operand is 0(Rn).

Example 5: Move the contents of the source address (contents of R4) to the destination (register R5). Register R4 is not modified.

Before operation: R4=A002h R5=050Ah Loc:0xA002=0123h

Operation: MOV @(R4), R5

After operation: R4= A002h R5=0123h Loc:0xA002=0123h

Operands	2 nd operand mode	Operator	Cycles	Length (words)
2	Register	Any	2*	1
2	Indexed, Symbolic or Absolute	Any	5	2
1	N/A	RRA, RRC, SWPB or SXT	3	1
1	N/A	CALL or PUSH	4	1

Table 7

6 Indirect auto increment mode

Similar to indirect register mode, but with indirect auto increment mode, the operand is incremented as part of the instruction. The format for operands is @Rn+. This is useful for working on blocks of data.

Description: Rn is used as a pointer to the operand. Rn is incremented afterwards by 1 for byte instructions and by 2 for word instructions.

Source mode bits: As = 11.

Syntax: @Rn+.

Length: One or two words.

Comment: Valid only for source operand. The substitute for destination operand is 0(Rn) plus second instruction INCD Rn.

Example 6: Move the contents of the source address (contents of R4) to the destination (register R5), then increment the value in register R4 to point to the next word.

Before operation: R4=A002h R5=050Ah Loc:0xA002=0123h

Operation: MOV @R4+, R5

After operation: R4= A004h R5=0123h Loc:0xA002=0123h

Operands	2 nd operand mode	Operator	Cycles	Length (words)
2	Register	Any	2*	1
2	Indexed, Symbolic or Absolute	Any	5	2
1	N/A	RRA, RRC, SWPB or SXT	3	1
1	N/A	PUSH	4	1
1	N/A	CALL	5	1

Table 8

7 Immediate mode

Immediate mode is used to assign constant values to registers or memory locations.

Description: The word following the instruction contains the immediate constant N. Indirect autoincrement mode @PC+ is used.

Source mode bits: As = 11.

Syntax: #N.

Length: Two or three words. It is one word less if a constant in CG1 or CG2 can be used.

Comment: Valid only for source operand.

Example 7: Move the immediate constant E2h to the destination (register R5).

Before operation: R4=A002h R5=050Ah

Operation: MOV #E2h, R5

After operation: R4= A002h R5=00E2h

Operands	2 nd operand mode	Operator	Cycles	Length (words)
2	Register	Any	2*	2
2	Indexed, Symbolic or Absolute	Any	5	3
1	N/A	RRA, RRC, SWPB or SXT	N/A	N/A
1	N/A	PUSH	4	2
1	N/A	CALL	5	2

Table 9

Request the MSP430 Teaching ROM Materials here https://www-a.ti.com/apps/dspuniv/teaching_rom_request.asp¹

¹https://www-a.ti.com/apps/dspuniv/teaching_rom_request.asp