1. Move 5 bytes of data to another location

; Source at 30H, destination at 40H

MOV R0, #30H

MOV R1, #40H

MOV R2, #05H

COPY\_LOOP:

MOV A, @R0

MOV @R1, A

INC R0

INC R1

DJNZ R2, COPY\_LOOP

1. Move 10 bytes of data to another location

; Source at 50H, destination at 60H

MOV R0, #50H

MOV R1, #60H

MOV R2, #0AH

COPY\_LOOP2:

MOV A, @R0

MOV @R1, A

INC R0

INC R1

DJNZ R2, COPY\_LOOP2

1. Move 4 bytes from one memory to another

; Source at 70H, destination at 80H

MOV R0, #70H

MOV R1, #80H

MOV R2, #04H

COPY\_LOOP3:

MOV A, @R0

MOV @R1, A

INC R0

INC R1

DJNZ R2, COPY\_LOOP3

1. Add and Subtract two 8-bit numbers using registers, display result in memory

; A = 25H, B = 15H

MOV A, #25H

MOV R0, #15H

; Addition

ADD A, R0

MOV 30H, A ; Store result at 30H

; Subtraction

MOV A, #25H

CLR C

SUBB A, R0

MOV 31H, A ; Store result at 31H

1. Add and Multiply two 8-bit numbers using registers, display result in memory

MOV A, #0FH

MOV B, #03H

; Addition

ADD A, B

MOV 40H, A ; Store addition result

; Multiplication

MOV A, #0FH

MOV B, #03H

MUL AB

MOV 41H, A ; Lower byte

MOV 42H, B ; Upper byte (in case result exceeds 8 bits)

1. Transfer 10 bytes from code memory to internal RAM

For this, you need to use MOVC instruction:

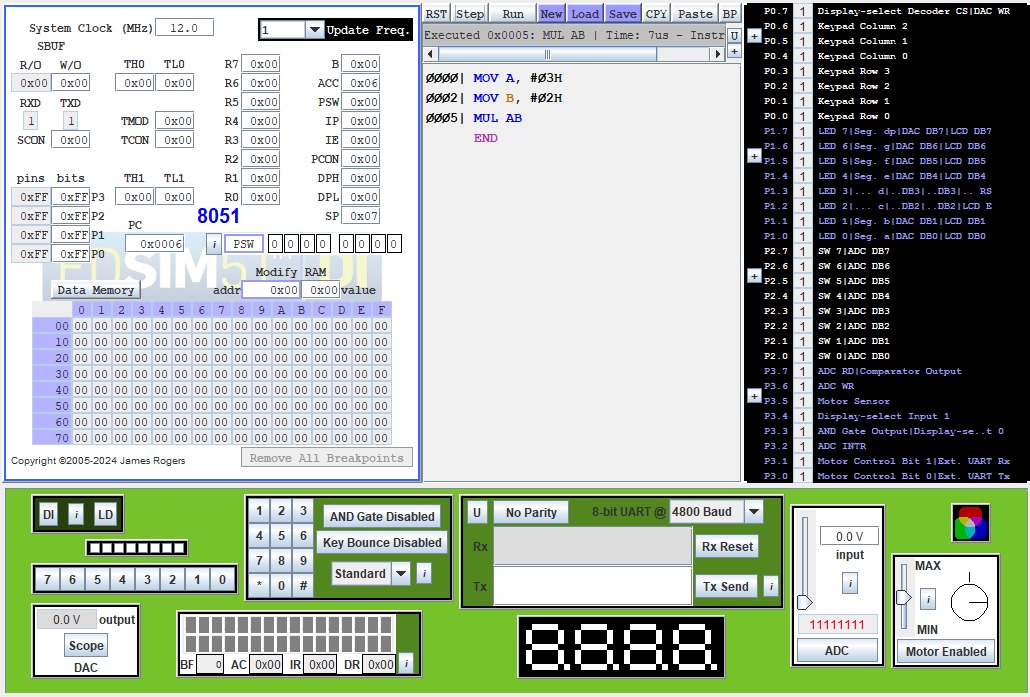
; Assume code memory starts at code label TABLE

; Destination internal RAM starts at 60H

MOV DPTR, #TABLE

MOV R1, #60H

MOV R2, #0AH



TRANSFER\_LOOP:

CLR A

MOVC A, @A+DPTR

MOV @R1, A

INC DPTR

INC R1

DJNZ R2, TRANSFER\_LOOP

SJMP $

TABLE:

DB 01H, 02H, 03H, 04H, 05H, 06H, 07H, 08H, 09H, 0AH

