

## Post Report 9

After doing experiment 9, I learned that Assembly language programming involves writing low-level code that is executed directly by the CPU. In assembly programming, data can be stored in registers or memory locations, and two commonly used methods for this are direct addressing and memory addressing.

Direct addressing is where storing data directly in registers, such as the al, bl, and ah registers, located within the CPU. It is known for its speed and efficiency, but registers have limited storage capacity, which makes it difficult to store large amounts of data.

```
AX=0005 BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=06B0 ES=06B0 SS=06B0 CS=06B0 IP=0106 NU UP EI PL NZ NA PO NC
06B0:0106 0408      ADD     AL,08
AX=000D BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=06B0 ES=06B0 SS=06B0 CS=06B0 IP=0108 NU UP EI PL NZ NA PO NC
06B0:0108 0404      ADD     AL,04
AX=0011 BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=06B0 ES=06B0 SS=06B0 CS=06B0 IP=010A NU UP EI PL NZ AC PE NC
06B0:010A 0407      ADD     AL,07
AX=0018 BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=06B0 ES=06B0 SS=06B0 CS=06B0 IP=010C NU UP EI PL NZ NA PE NC
06B0:010C 0406      ADD     AL,06
AX=001E BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=06B0 ES=06B0 SS=06B0 CS=06B0 IP=010E NU UP EI PL NZ NA PE NC
06B0:010E B305      MOV     BL,05
AX=001E BX=0005 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=06B0 ES=06B0 SS=06B0 CS=06B0 IP=0110 NU UP EI PL NZ NA PE NC
06B0:0110 F6F3      DIV     BL
AX=0006 BX=0005 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=06B0 ES=06B0 SS=06B0 CS=06B0 IP=0112 NU UP EI PL NZ NA PE NC
06B0:0112 0000      ADD     [BX+SI],AL      DS:0005=EA
```

Memory addressing on the other hand involves storing data in memory locations that are accessed by the CPU. It provides more flexibility and ease of modification but is generally less efficient due to the need for additional commands to load and store data.

```
AX=0005 BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=06B0 ES=06B0 SS=06B0 CS=06B0 IP=0107 NU UP EI PL NZ NA PO NC
06B0:0107 02060102      ADD     AL,[0201]      DS:0201=08
AX=000D BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=06B0 ES=06B0 SS=06B0 CS=06B0 IP=0108 NU UP EI PL NZ NA PO NC
06B0:0108 02060202      ADD     AL,[0202]      DS:0202=04
AX=0011 BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=06B0 ES=06B0 SS=06B0 CS=06B0 IP=010F NU UP EI PL NZ AC PE NC
06B0:010F 02060302      ADD     AL,[0203]      DS:0203=07
AX=0018 BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=06B0 ES=06B0 SS=06B0 CS=06B0 IP=0113 NU UP EI PL NZ NA PE NC
06B0:0113 02060402      ADD     AL,[0204]      DS:0204=06
AX=001E BX=0000 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=06B0 ES=06B0 SS=06B0 CS=06B0 IP=0117 NU UP EI PL NZ NA PE NC
06B0:0117 B305      MOV     BL,05
AX=001E BX=0005 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=06B0 ES=06B0 SS=06B0 CS=06B0 IP=0119 NU UP EI PL NZ NA PE NC
06B0:0119 F6F3      DIV     BL
AX=0006 BX=0005 CX=0000 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=06B0 ES=06B0 SS=06B0 CS=06B0 IP=011B NU UP EI PL NZ NA PE NC
06B0:011B 0000      ADD     [BX+SI],AL      DS:0005=EA
```

The choice between the two methods depends on the specific requirements of the program. If speed and efficiency are crucial, direct addressing may be the best option. If flexibility and ease of modification are more important, memory addressing may be preferred. Understanding the differences between these two methods is essential for writing efficient and effective assembly language programs. Programmers need to consider the requirements of their program to decide which method to use for optimized performance.