SOLUTIONS FOR CHAPTER 1

1. 8086

2. the internal data bus of the 386SX is 32 bits, whereas the internal data bus of the 286 is 16 bits

3. terms such as "16-bit" or "32-bit" microprocessors refer to the internal data bus and register size of the microprocessor

4. yes

5. upward compatibility means that any program written for a lower (earlier) system will run on more advanced (later) systems

6. the 8088 has an 8-bit external data bus but the 8086 has a 16-bit external data bus

7. the 8088 has a 4-byte queue, the 8086 has a 6-byte queue

8. more efficient internal architecture such as pipelining and wider registers

9. the BIU (bus interface unit) fetches instructions into the CPU and the EU (execution unit) executes the instruction

10. (a) 8-bit registers are: AH, AL, BH, BL, CH, CL, DH, CL
    (b) 16-bit registers are: AX, BX, CX, DX

11. (a) CS (c) DS (d) SS (h) SI (i) DI

12. (b) is illegal since the value is too large
    (c) is illegal since immediate addressing is not allowed for segment registers
    (f) is illegal since immediate addressing is not allowed for segment registers
    (i) is illegal since the operand types do not match
    (j) is illegal since the value is too large for the register
    (k) is illegal since the register sizes do not match
    (l) is illegal since the operand sizes do not match

13. CS is the code segment register and holds the segment address for the code section
    DS is the data segment register and holds the segment address for the data section
    SS is the stack segment register and holds the segment address for the stack section
    ES is the extra segment register and holds the segment address for the extra segment which is used for many string operations

14. (a) 3499:2500 (b) 36E90 (c) 34990 to 4498F

15. (a) 1296:0100 (b) 12A60 (c) 12960 to 2295F

16. (a) 38949 (b) 3499:3FB9 (c) 34990 to 4498F

17. (a) 1A648 (b) 1298:7CC8 (c) 12980 to 2297F

18. 0042:004C

19. no, because the upper range of the code segment would be 36FFF
    CS should be 3777

20.

12B0:0170	12C70	B0
12B0:0171	12C71	76
12B0:0172	12C72	B7
12B0:0173	12C73	8F
12B0:0174	12C74	00
12B0:0175	12C75	C7
12B0:0176	12C76	80
12B0:0177	12C77	C7
12B0:0178	12C78	7B
12B0:0179	12C79	88
12B0:017A	12C7A	FB
12B0:017B	12C7B	00
12B0:017C	12C7C	C3

21.

12B0:0100	12C00	B0
12B0:0101	12C01	00
12B0:0102	12C02	02
12B0:0103	12C03	06
12B0:0104	12C04	00
12B0:0105	12C05	02
12B0:0106	12C06	02
12B0:0107	12C07	06
12B0:0108	12C08	01
12B0:0109	12C09	02
12B0:010A	12C0A	02
12B0:010B	12C0B	06
12B0:010C	12C0C	02
12B0:010D	12C0D	02
12B0:010E	12C0E	02
12B0:010F	12C0F	06
12B0:0110	12C10	03
12B0:0111	12C11	02
12B0:0112	12C12	02
12B0:0113	12C13	06
12B0:0114	12C14	04
12B0:0115	12C15	02

22. (b)

23. (c)

24. decremented, incremented

25. (b)

26. the stack is slower than registers, since the stack is a section of RAM

27. (a) 24578 (b) 2000:4578 (c) 20000 (d) 2FFFF

28. 24FB

29. after "PUSH AX", the stack pointer = FF2C and the stack is as follows:

logical address	stack contents
SS:FF2C	91
SS:FF2D	32

after "PUSH BX", the stack pointer = FF2A and the stack is as follows:

logical address	stack contents
SS:FF2A	3C
SS:FF2B	F4
SS:FF2C	91
SS:FF2D	32

after "PUSH CX", the stack pointer = FF28 and the stack is as follows:

logical address	stack contents
SS:FF28	09
SS:FF29	00
SS:FF2A	3C
SS:FF2B	F4
SS:FF2C	91
SS:FF2D	32

30. at the conclusion of Problem 28, SP = FF28,
```
POP CX      ;then SP = FF2A
POP BX      ;then SP = FF2C
POP AX      ;then SP = FF2E
```

31. (a) SS (b) DS (c) CS (d) DS
(e) SS (f) DS

32. (a) SS overrides default register DS
(b) SS overrides default register DS
(c) DS overrides default register SS

33. (a) CF = 1 indicating a carry occurred
PF = 1 indicating even parity
AF = 1 indicating a carry from bit 3
ZF = 1 indicating the result is zero
SF = 0 indicating a positive result

(b) CF = 0 indicating no carry
PF = 0 indicating odd parity
AF = 0 indicating no carry from bit 3
ZF = 0 indicating that the result is not zero
SF = 1 indicating negative result

(c) CF = 0 indicating no carry
PF = 1 indicating even parity
AF = 1 indicating a carry from bit 3
ZF = 0 indicating the result is not zero
SF = 0 indicating positive result

34. (a) location 24000 (20000 + 4000) contains FF
(b) location 2A088 (20000 + 4000 + 6080 + 8) contains 25
(c) location 26080 (20000 + 6080) contains FF
location 26081 contains 25
(d) location 25006 (20000 + 5000 + 6) contains 80
location 25007 contains 60
(e) location 2B0A8 (20000 + 5000 + 6080 + 28) contains 91
location 2B0A9 contains 87
(f) location 34010 (30000 + 4000 + 10) contains 99
location 34011 contains 12

(g) location 23600 (20000 + 3600) contains FF
location 23601 contains 25
(h) location 260B0 (20000 + 6080 + 30) contains 99
location 260B1 contains 12
(i) location 37200 (30000 + 7000 + 200) contains FF
location 37201 contains 25
(j) location 3B100 (30000 + 7000 + 4000 + 100) contains 80
location 3B101 contains 60
(k) location 24050 (20000 + 4000 + 50) contains 25
(l) location 2C100 (20000 + 5000 + 7000 + 100) contains FF
location 2C101 contains 25

35. (a) register
(b) immediate
(c) direct
(d) register
(e) register indirect
(f) register indirect
(g) based index
(h) register
(i) based
(j) based index
(k) index
(l) based index

36. (a) DS:1450 contains 9F
DS:1451 contains 12
(b) DS:2348 contains 63
DS:2349 contains 8C