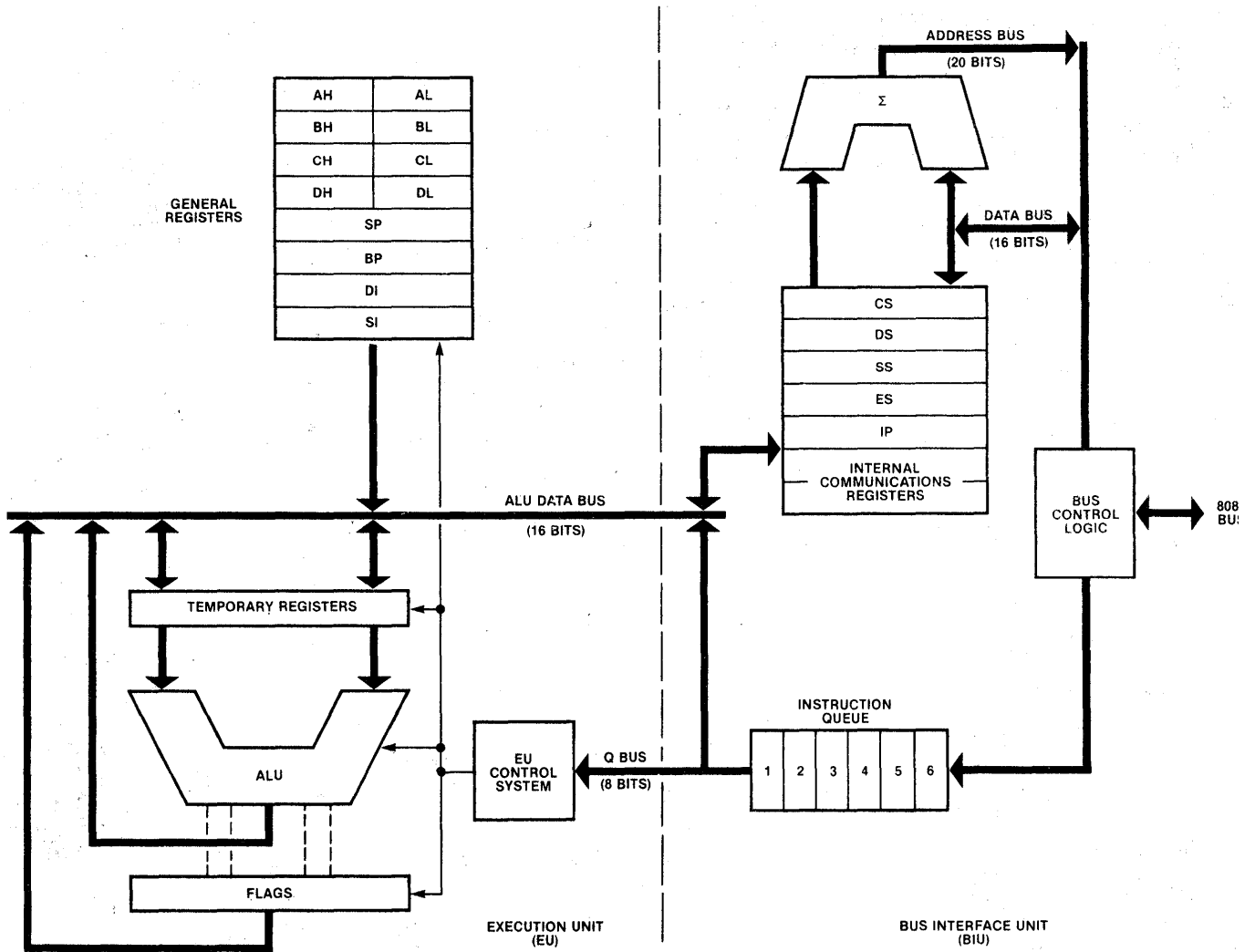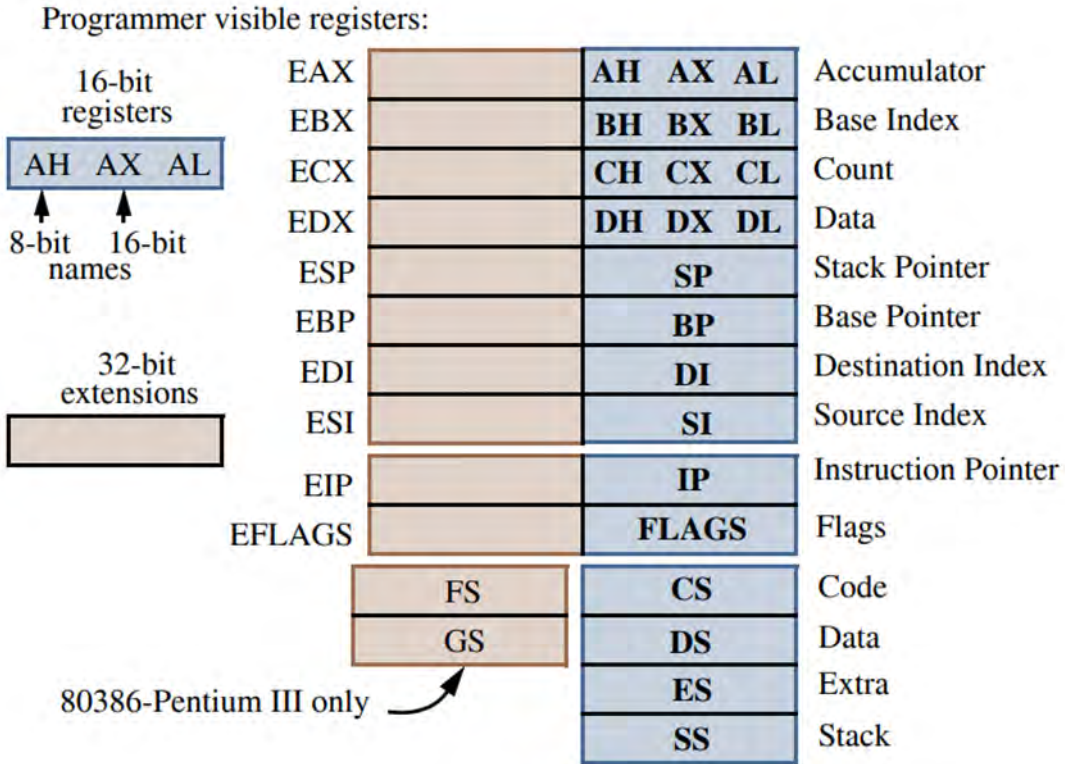Please return this document along with your exam sheet at the end of your exam
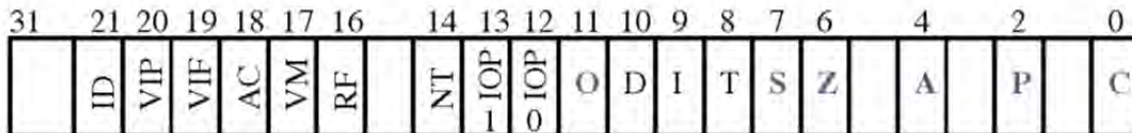
8086 Architecture

General-Purpose Registers - 32 bit mode

Programmer visible registers:

16-bit registers			
	EAX	AH AX AL	Accumulator
	EBX	BH BX BL	Base Index
AH AX AL	ECX	CH CX CL	Count
↑ ↑	EDX	DH DX DL	Data
8-bit 16-bit names	ESP	SP	Stack Pointer
	EBP	BP	Base Pointer
32-bit extensions	EDI	DI	Destination Index
	ESI	SI	Source Index
	EIP	IP	Instruction Pointer
	EFLAGS	FLAGS	Flags
	FS	CS	Code
	GS	DS	Data
80386-Pentium III only		ES	Extra
		SS	Stack

EFLAGs Register

31		21	20	19	18	17	16		14	13	12	11	10	9	8	7	6		4		2		0
		ID	VIP	VIF	AC	VM	RF		NT	IOP 1	IOP 0	O	D	I	T	S	Z		A		P		C

For the table below, definitions for flag abbreviations are

OF - Overflow Flag, DF - Direction Flag, SF - Sign Flag, ZF - Zero Flag, AF - Auxiliary Flag
PF - Parity Flag, CF - Carry Flag

A ✓ means, the flag is set as per the outcome of the result, 0 - clear and 1 - set

Instruction	Brief Description	Flags Affected						
		OF	DF	SF	ZF	AF	PF	CF
MOV	Move data between: registers, memory and registers, immediate values to registers							
MOVD	Move doubleword							
ADD	Integer Addition	✓		✓	✓	✓	✓	✓
SUB	Integer Subtraction	✓		✓	✓	✓	✓	✓
MUL	Unsigned Multiply	✓						✓
IMUL	Signed Multiply	✓						✓
DIV	Unsigned Divide							
IDIV	Signed Divide							
INC	Increment	✓		✓	✓	✓	✓	
DEC	Decrement	✓		✓	✓	✓	✓	
NEG	Negate	✓		✓	✓	✓	✓	✓
CMP	Comparison	✓		✓	✓	✓	✓	✓
AND	Bitwise Logical AND	0		✓	✓		✓	0
OR	Bitwise Logical OR	0		✓	✓		✓	0
XOR	Bitwise Logical XOR	0		✓	✓		✓	0
NOT	Bitwise Logical NOT							
SHR	Shift Right	✓		✓	✓		✓	✓
SHL	Shift Left	✓		✓	✓		✓	✓
ROR	Rotate Right	✓						✓
ROL	Rotate Left	✓						✓
RCR	Rotate Through Carry Right	✓						✓
RCL	Rotate Through Carry Left	✓						✓

For the table below, definitions for flag abbreviations are

OF - Overflow Flag, DF - Direction Flag, SF - Sign Flag, ZF - Zero Flag, AF - Auxiliary Flag
PF - Parity Flag, CF - Carry Flag

A ✓ means, the flag is set as per the outcome of the result, 0 - clear and 1 - set

Instruction	Brief Description	Flags Affected						
		OF	DF	SF	ZF	AF	PF	CF
LEA	Load Effective Address							
NOP	No Operation							
LOOP	Loop with ECX counter							
LOOPZ/LOOPE	Loop with ECX and zero/ Loop with ECX and equal							
LOOPNZ/ LOOPNE	Loop with ECX and not zero/ Loop with ECX and not equal							
PUSH	Push onto stack							
POP	Pop off stack							
CALL	Call procedure	✔	✔	✔	✔	✔	✔	✔
RET	Return from procedure							
LODS/LODB/ LODSW/LODSD	Load string/ load byte string Load word string Load double word string							
STOS/STOB/STOSW/STOSD	Store string/ Store byte string Store word string Store double word string							
SCAS/SCASB/SCASW/SCASD	Scan string Scan byte string Scan word string Scan double word string							
CLD	Clear Direction Flag (Auto-Increment RSI/RDI)							
STD	Set Direction Flag (Auto-Decrement RSI/RDI)							
REP	Repeat while ECX not zero							
REPE/REPZ	Repeat while Equal/ Repeat while zero							
REPNE/REPNZ	Repeat while Not Equal/ Repeat while Not zero							

Instruction	Brief Description	Flags Affected							
JMP	Unconditional Jump								

Conditional jumps (Jcc)

Jumps to relative address based on the state of flags

Instruction Mnemonic	Condition (Flag States)	Description
Unsigned Conditional Jumps		
JA/JNBE JAE/JNB JB/JNAE JBE/JNA JC JE/JZ JNC JNE/JNZ JNP/JPO JP/JPE JCXZ JECXZ	(CF and ZF) = 0 CF = 0 CF= 1 (CF or ZF) = 1 CF = 1 ZF = 1 CF = 0 ZF = 0 PF = 0 PF = 1 CX = 0 ECX = 0	Above/not below or equal Above or equal/not below Below/not above or equal Below or equal/not above Carry Equal/zero Not Carry Not equal/not zero Not parity/parity odd Parity/parity even Register CX is zero Register ECX is zero
Signed Conditional Jumps		
JG/JNLE JGE/JNL JL/JNGE JLE/JNG JNO JNS JO JS	SF = OF, and ZF = 0 SF = OF SF ≠ OF SF ≠ OF, or ZF = 1 OF = 0 SF = 0 OF = 1 SF = 1	Greater/not less or equal Greater or equal/not less Less/not greater or equal Less or equal/not greater Not overflow Not sign (non-negative) Overflow Sign (negative)

Declaring Initialized Data

DB, DW, DD, DQ, DT (80-bit)

Instruction	Operand	Comment
db	0x55	; just the byte 0x55
db	0x55,0x56,0x57	; three bytes in succession
db	'a',0x55	; character constants are OK
db	'hello',13,10,'$'	; so are string constants
dw	0x1234	; 0x34 0x12
dw	'a'	; 0x61 0x00 (it's just a number)
dw	'ab'	; 0x61 0x62 (character constant)
dw	'abc'	; 0x61 0x62 0x63 0x00 (string)
dd	0x12345678	; 0x78 0x56 0x34 0x12
dd	1.234567E+20	; floating-point constant
dq	0x123456789abcdef0	; eight byte constant
dq	1.234567E+20	; double-precision float
dt	1.234567E+20	; extended-precision float

Reserving Uninitialized Data

RESB, RESW, RESD, RESQ, REST

Label	Instruction	Operand	Comment
buffer:	resb	64	; reserve 64 bytes
wordvar:	resw	1	; reserve a word
realarray:	resq	10	; array of ten reals