SOFTWARE ARCHITECTURE OF THE 8088 AND 8086 MICROPROCESSORS

SOFTWARE ARCHITECTURE OF THE 8088 AND 8086 MICROPROCESSORS

2.1 Microarchitecture of the 8088/8086 Microprocessor
2.2 Software Model of the 8088/8086 Microprocessor
2.3 Memory Address Space and Data Organization
2.4 Data Types
2.5 Segment Registers and Memory Segmentation
2.6 Dedicated, Reserved, and General-Used Memory
2.7 Instruction Pointer

國立台灣大學
生物機電系
林達德

SOFTWARE ARCHITECTURE OF THE 8088 AND 8086 MICROPROCESSORS

2.8 Data Registers
2.9 Pointer and Index Register
2.10 Status Register
2.11 Generating a Memory Address
2.12 The Stack
2.13 Input/Output Address Space

2.1 Microarchitecture of the 8088/8086 Microprocessor

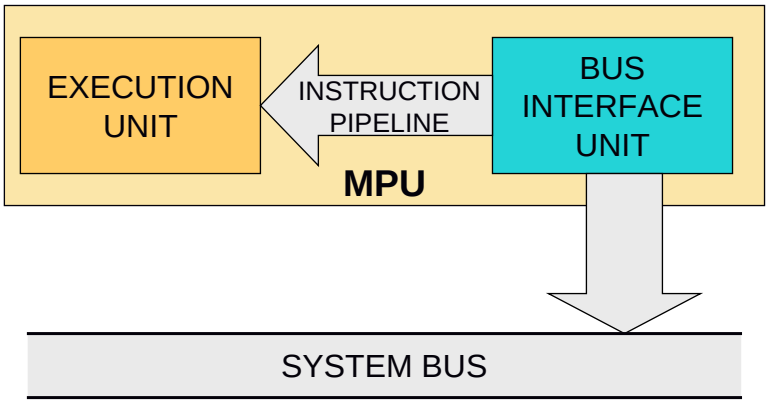
- 8088/8086 both employ ***parallel processing***
- 8088/8086 contain two processing unit – the ***bus interface unit*** (BIU) and ***execution unit*** (EU)
- The bus interface unit is the path that 8088/8086 connects to external devices.
- The system bus includes an 8-bit bidirectional data bus for 8088 (16 bits for the 8086), a 20-bit address bus, and the signal needed to control transfers over the bus.

2.1 Microarchitecture of the 8088/8086 Microprocessor

- Components in BIU
  - Segment register
  - The instruction pointer
  - Address generation adder
  - Bus control logic
  - Instruction queue
- Components in EU
  - Arithmetic logic unit, ALU
  - Status and control flags
  - General-purpose registers
  - Temporary-operand registers

國立台灣大學
生物機電系
林達德

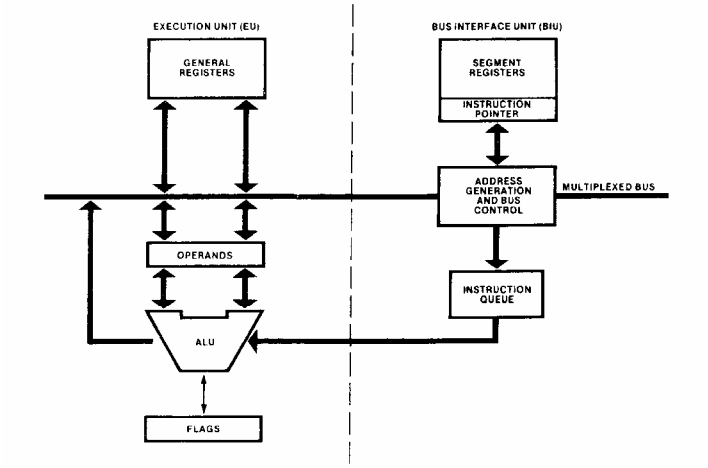
2.1 Microarchitecture of the 8088/8086 Microprocessor



Pipeline architecture of the 8086/8088 microprocessors

國立台灣大學
生物機電系
林達德

2.1 Microarchitecture of the 8088/8086 Microprocessor

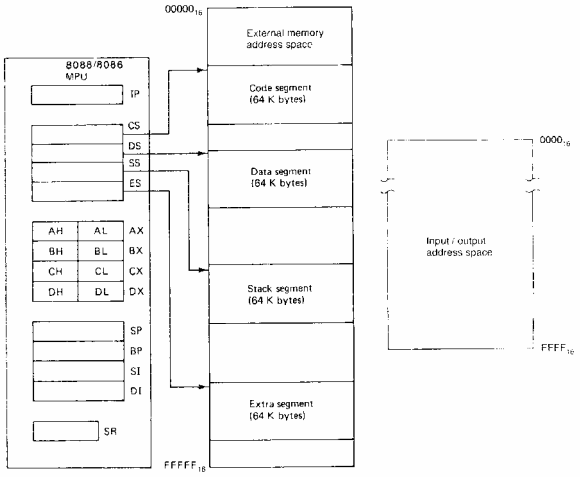


EU and BIU of the 8086/8088 microprocessors

2.2 Software Model of the 8088/8086 Microprocessor

- 8088 microprocessor includes 13 16-bit internal registers.
  - The instruction pointer, IP
  - Four data registers, AX, BX, CX, DX
  - Two pointer register, BP, SP
  - Two index register, SI, DI
  - Four segment registers, CS, DS, SS, ES
- The status register, SR, with nine of its bits implement for status and control flags.
- The memory address space is 1 Mbytes and the I/O address space is 64 Kbytes in length.

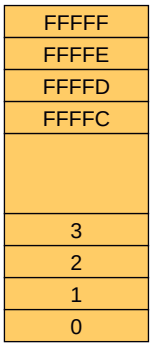
2.2 Software Model of the 8088/8086 Microprocessor



Software model of the 8088/8086 microprocessors

2.3 Memory Address Space and Data Organization

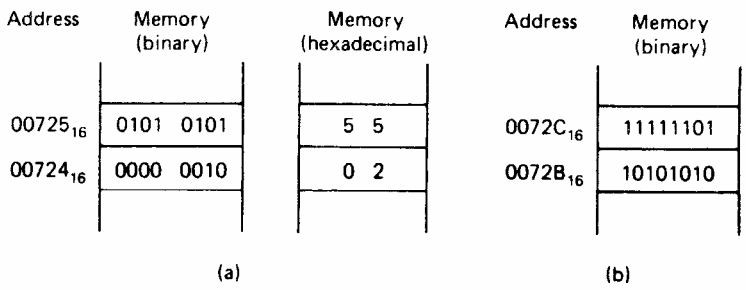
- The 8088 microcomputer supports 1 Mbytes of external memory.
- The memory of an 8088-based microcomputer is organized as 8-bit bytes, not as 16-bit words.



Memory address space of the 8088/8086 microcomputer

2.3 Memory Address Space and Data Organization

- Lower address byte and higher address byte



The two bytes represent the word
$0101010100000010_2 = 5502_{16}$

國立台灣大學
生物機電系
林達德

2.3 Memory Address Space and Data Organization

EXAMPLE

What is the data word shown in the previous figure? Express the result in hexadecimal form. Is it stored at an even- or odd-addressed word boundary? Is it an aligned or misaligned word of data?

Solution:

$11111101_2 = FD_{16} = FDH$

$10101010_2 = AA_{16} = AAH$

Together the two bytes give the word

$1111110110101010_2 = FDAA_{16} = FDAAH$

Expressing the address of the least significant byte in binary form gives $0072BH = 0072B_{16} = 00000000011100101011_2$

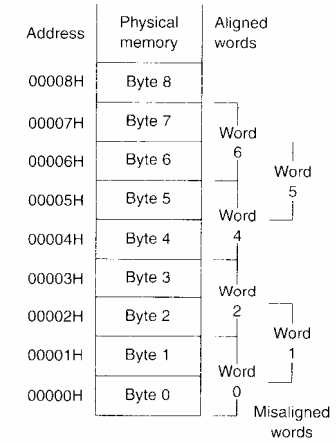
Therefore, it is misaligned word of data.

國立台灣大學
生物機電系
林達德

2.3 Memory Address Space and Data Organization

- Even- or odd-addressed word

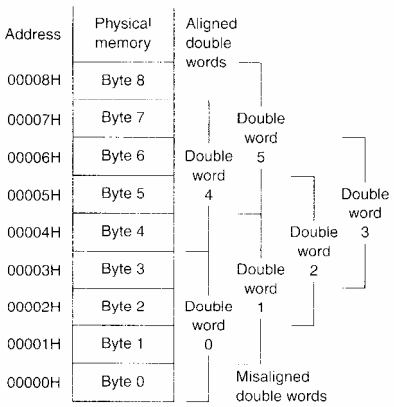
  If the least significant bit of the address is 0, the word is said to be held at an even-addressed boundary.
- Aligned word or misaligned word



國立台灣大學
生物機電系
林達德

2.3 Memory Address Space and Data Organization

- A ***double word*** corresponds to four consecutive bytes of data stored in memory.



國立台灣大學
生物機電系
林達德

2.3 Memory Address Space and Data Organization

- A pointer is a double word. The higher address word represents the ***segment base address*** while the lower address word represents the ***offset*** .

Address	Memory (binary)	Memory (hexadecimal)	Address	Memory (hexadecimal)
00007_{16}	0011 1011	3 B	$0000B_{16}$	A0
00006_{16}	0100 1100	4 C	$0000A_{16}$	00
00005_{16}	0000 0000	0 0	00009_{16}	55
00004_{16}	0110 0101	6 5	00008_{16}	FF

Example: Segment base address = $3B4C_{16} = 0011101101001100_2$
Offset value = $0065_{16} = 0000000001100101_2$

國立台灣大學
生物機電系
林達德

2.3 Memory Address Space and Data Organization

EXAMPLE

How should the pointer with segment base address equal to $A000_{16}$ and offset address $55FF_{16}$ be stored at an even-address boundary starting at 00008_{16}? Is the double word aligned or misaligned?

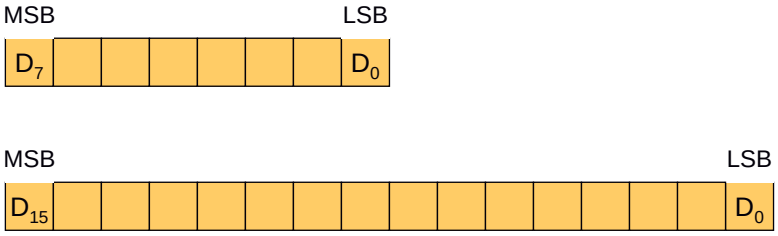
Solution:

Storage of the two-word pointer requires four consecutive byte locations in memory, starting at address 00008_{16}. The least-significant byte of the offset is stored at address 00008_{16} and is shown as FF_{16} in the previous figure. The most significant byte of the offset, 55_{16}, is stored at address 00009_{16}. These two bytes are followed by the least significant byte of the segment base address, 00_{16}, at address $0000A_{16}$, and its most significant byte, $A0_{16}$, at address $0000B_{16}$. Since the double word is stored in memory starting at address 00008_{16}, it is aligned.

國立台灣大學
生物機電系
林達德

2.4 Data Types

- Integer data type
  - Unsigned or signed integer
  - Byte-wide or word-wide integer

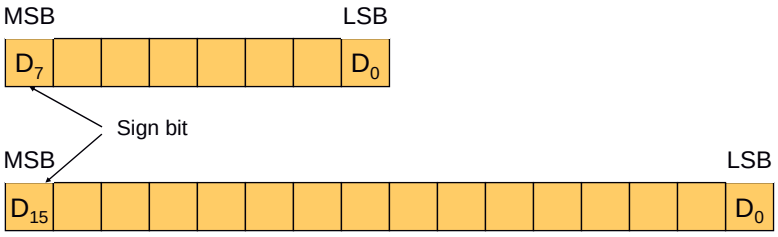


Unsigned byte and unsigned word integer

國立台灣大學
生物機電系
林達德

2.4 Data Types

- The most significant bit of a signed integer is a ***sign bit***. A zero in this bit position identifies a positive number.
- The range of a signed byte integer is +127 ~ -128. The range of a signed word integer is +32767 ~ -32768.
- The 8088 always expresses negative numbers in 2's-complement.



Signed byte and signed word integer

國立台灣大學
生物機電系
林達德

2.4 Data Types

EXAMPLE

A signed word integer equals $FEFF_{16}$. What decimal number does it represent?

Solution:

$$FEFF_{16} = 1111111011111111_2$$

The most significant bit is 1, the number is negative and is in 2's complement form.

Converting to its binary equivalent by subtracting 1 from the least significant bit and then complement all bits give

$$FEFF_{16} = -0000000100000001_2$$
$$= -257$$

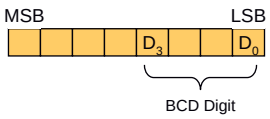
國立台灣大學
生物機電系
林達德

2.4 Data Types

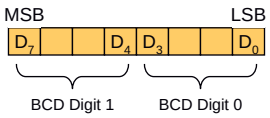
- The 8088 can also process data that is coded as ***binary-coded decimal (BCD) numbers.***
- BCD data can be stored in packed or unpacked forms.

Decimal	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

BCD Numbers



Unpacked BCD digit



Packed BCD digit

國立台灣大學
生物機電系
林達德

2.4 Data Types

EXAMPLE

The packed BCD data stored at byte address 01000_{16} equals 10010001_2. What is the two digit decimal number?

Solution:

Writing the value 10010001_2 as separate BCD digits gives

$$10010001_2 = 1001_{BCD}0001_{BCD} = 91_{10}$$

國立台灣大學
生物機電系
林達德

2.4 Data Types

- The ASCII (American Standard Code for Information Interchange) digit

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

國立台灣大學
生物機電系
林達德

2.4 Data Types

EXAMPLE

Byte addresses 01100_{16} through 01104_{16} contain the ASCII data 01000001, 01010011, 01000011, 01001001, and 01001001, respectively. What do the data stand for?

Solution:

Using the ASCII table, the data are converted to ASCII code:

(01100H) = 01000001_2 = A
(01101H) = 01010011_2 = S
(01102H) = 01000011_2 = C
(01103H) = 01001001_2 = I
(01104H) = 01001001_2 = I

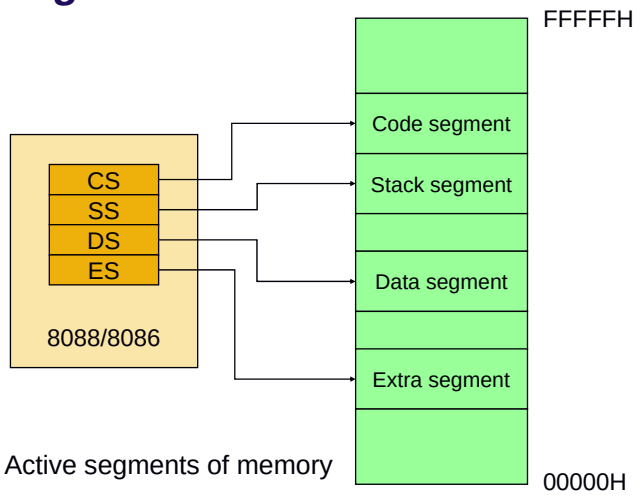
國立台灣大學
生物機電系
林達德

2.5 Segment Registers and Memory Segmentation

- A ***segment*** represents an independently addressable unit of memory consisting of 64K consecutive byte-wide storage locations.
- Each segment is assigned a ***base address*** that identifies its starting point.
- Only four segments can be active at a time:
  - The code segment
  - The stack segment
  - The data segment
  - The extra segment
- The addresses of the active segments are stored in the four internal segment registers: CS, SS, DS, ES.

國立台灣大學
生物機電系
林達德

2.5 Segment Registers and Memory Segmentation



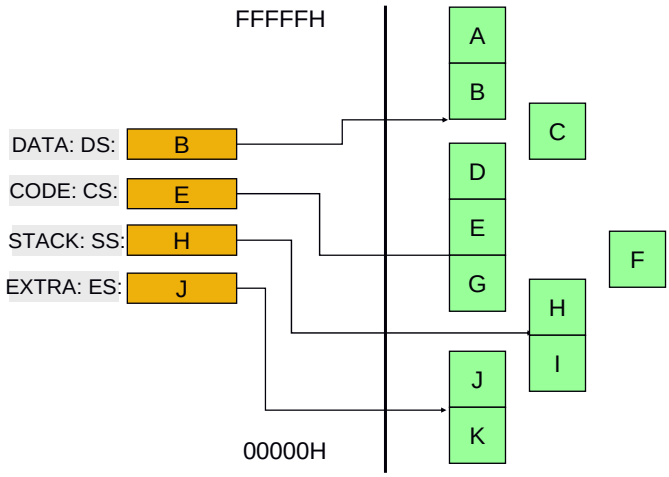
國立台灣大學
生物機電系
林達德

2.5 Segment Registers and Memory Segmentation

- Four segments give a maximum of 256Kbytes of active memory.
  - Code segment – 64K
  - Stack – 64K
  - Data storage – 128K
- The base address of a segment must reside on a 16-byte address boundary.
- User accessible segments can be set up to be contiguous, adjacent, disjointed, or even overlapping.

國立台灣大學
生物機電系
林達德

2.5 Segment Registers and Memory Segmentation



Contiguous, adjacent, disjointed, and overlapping segments

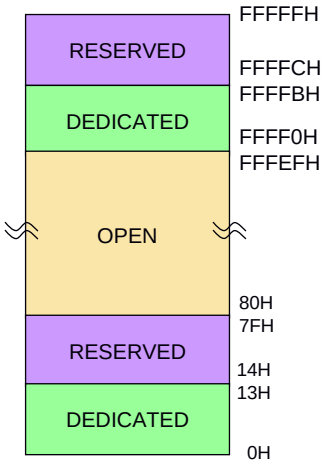
國立台灣大學
生物機電系
林達德

2.6 Dedicated, Reserved, and General-Used Memory

- The ***dedicated memory*** (00000_{16} ~ 00013_{16}) are used for storage of the pointers to 8088's internal interrupt service routines and exceptions.
- The ***reserved memory*** (00014_{16} ~ $0007F_{16}$) are used for storage of the pointers to user-defined interrupts.
- The 128-byte dedicated and reserved memory can contain 32 interrupt pointers.
- The ***general-use memory*** (00080_{16} ~ $FFFEF_{16}$) stores data or instructions of the program.
- The dedicated memory ($FFFE0_{16}$ ~ $FFFEB_{16}$) are used for hardware reset jump instruction.

國立台灣大學
生物機電系
林達德

2.6 Dedicated, Reserved, and General-Used Memory



國立台灣大學
生物機電系
林達德

2.7 Instruction Pointer

- The ***instruction pointer (IP)*** identifies the location of the next word of instruction code to be fetched from the current code segment of memory.
- The offset in IP is combined with the current value in CS to generate the address of the instruction code.
- During normal operation, the 8088 fetches instructions from the code segment of memory, stores them in its instruction queue, and executes them one after the other.

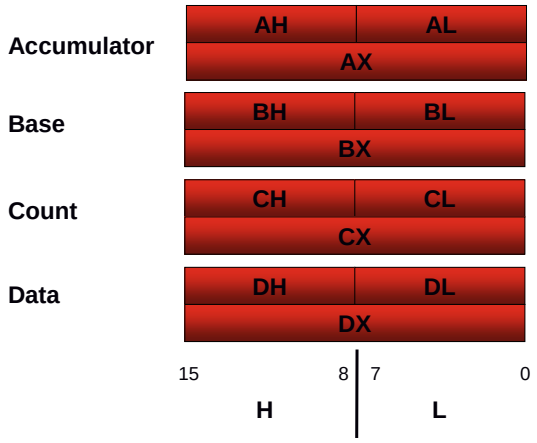
國立台灣大學
生物機電系
林達德

2.8 Data Registers

- Data registers are used for temporary storage of frequently used intermediate results.
- The contents of the data registers can be read, loaded, or modified through software.
- The four data registers are:
  - Accumulator register, A
  - Base register, B
  - Counter register, C
  - Data register, D
- Each register can be accessed either as a whole (16 bits) for word data or as 8-bit data for byte-wide operation.

國立台灣大學
生物機電系
林達德

2.8 Data Registers



General-purpose data registers of 8088 microprocessor

國立台灣大學
生物機電系
林達德

2.8 Data Registers

Register	Operations
AX	Word multiply, word divide, word I/O
AL	Byte multiply, byte divide, byte I/O, translate, decimal arithmetic
AH	Byte multiply, byte divide
BX	Translate
CX	String operations, loops
CL	Variable shift and rotate
DX	Word multiply, word divide, indirect I/O

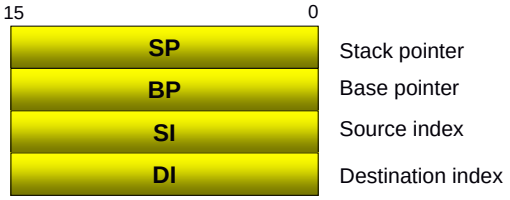
Dedicated register functions

2.9 Pointer and Index Register

- The pointer registers and index registers are used to store offset addresses.
- Values held in the index registers are used to reference data relative to the data segment or extra segment.
- The pointer registers are used to store offset addresses of memory location relative to the stack segment register.
- Combining SP with the value in in SS (SS:SP) results in a 20-bit address that points to the ***top of the stack*** (TOS).
- BP is used to access data within the stack segment of memory. It is commonly used to reference subroutine parameters.

2.9 Pointer and Index Register

- The index register are used to hold offset addresses for instructions that access data in the data segment.
- The source index register (SI) is used for a source operand, and the destination index (DI) is used for a destination operand.
- The four registers must always be used for 16-bit operations.



國立台灣大學
生物機電系
林達德

2.10 Status Register

- The status register, also called the flags register, indicate conditions that are produced as the result of executing an instruction.
- Only nine bits of the register are implemented. Six of these bits represent ***status flags*** and the other three bits represent ***control flags***
- The 8088 provides instructions within its instruction set that are able to use these flags to alter the sequence in which the program is executed.

國立台灣大學
生物機電系
林達德

2.10 Status Register

Status and control bits maintained in the flags register

15															0
x	x	x	x	OF	DF	IF	TF	SF	ZF	x	AF	x	PF	x	CF

- Generally Set and Tested Individually
- 9 1-bit flags in 8086; 7 are unused

國立台灣大學
生物機電系
林達德

2.10 Status Register

- Status flags indicate current processor status.

CF	Carry Flag	Arithmetic Carry/Borrow
OF	Overflow Flag	Arithmetic Overflow
ZF	Zero Flag	Zero Result; Equal Compare
SF	Sign Flag	Negative Result; Non-Equal Compare
PF	Parity Flag	Even Number of “1” bits
AF	Auxiliary Carry	Used with BCD Arithmetic

國立台灣大學
生物機電系
林達德

2.10 Status Register

- Control flags influence the 8086 during execution phase

DF	Direction Flag	Auto-Increment/Decrement
	used for “string operations”	
IF	Interrupt Flag	Enables Interrupts
	allows “fetch-execute” to be interrupted	
TF	Trap Flag	Allows Single-Step
	for debugging; causes interrupt after each op	

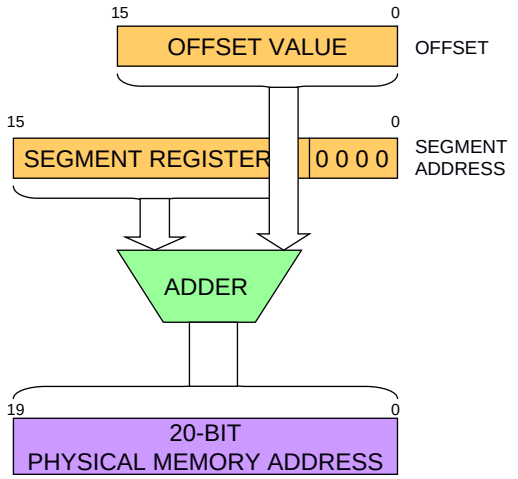
國立台灣大學
生物機電系
林達德

2.11 Generating a Memory Address

- A ***logical address*** in the 8088 microcomputer system is described by a segment base and an offset.
- The ***physical addresses*** that are used to access memory are 20 bits in length.
- The generation of the physical address involves combining a 16-bit offset value that is located in the instruction pointer, a base pointer, an index register, or a pointer register and a 16-bit segment base value that is located in one of the segment register.

國立台灣大學
生物機電系
林達德

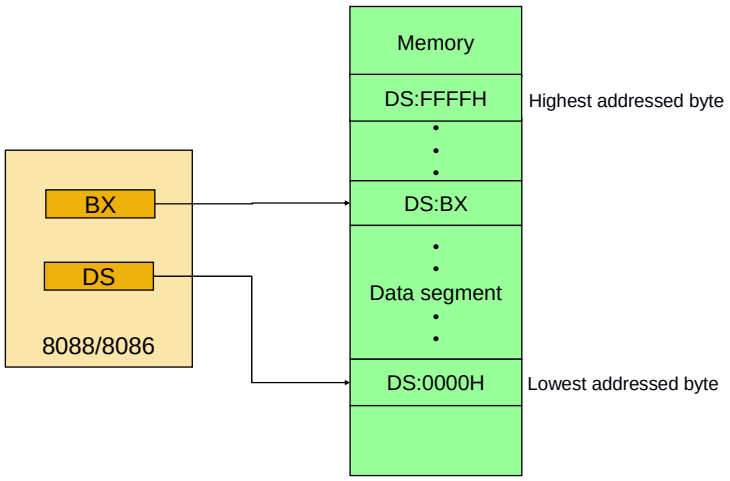
2.11 Generating a Memory Address



Generating a physical address

國立台灣大學
生物機電系
林達德

2.11 Generating a Memory Address

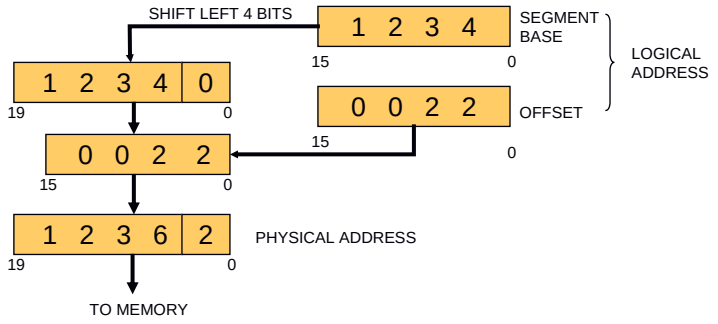


Boundary of a segment

國立台灣大學
生物機電系
林達德

2.11 Generating a Memory Address

EXAMPLE



國立台灣大學
生物機電系
林達德

2.11 Generating a Memory Address

EXAMPLE

What would be the offset required to map to physical address location $002C3_{16}$ if the contents of the corresponding segment register are $002A_{16}$?

Solution:

The offset value can be obtained by shifting the contents of the segment of the segment register left by four bit positions and then subtracting from the physical address. Shifting left give

$$002A0_{16}$$

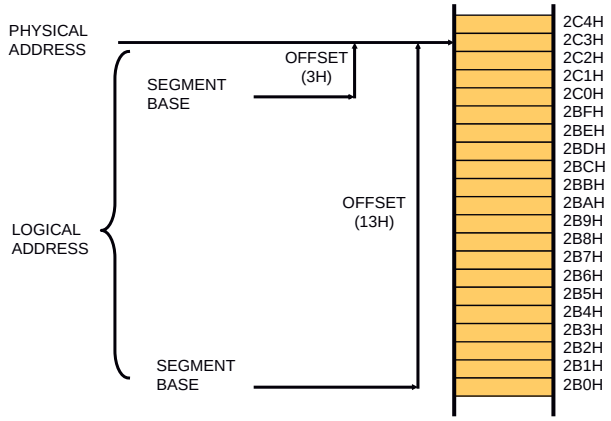
Now subtracting, we get the value of the offset:

$$002C3_{16} - 002A0_{16} = 0023_{16}$$

國立台灣大學
生物機電系
林達德

2.11 Generating a Memory Address

- Different logical addresses can be mapped to the same physical address location in memory.



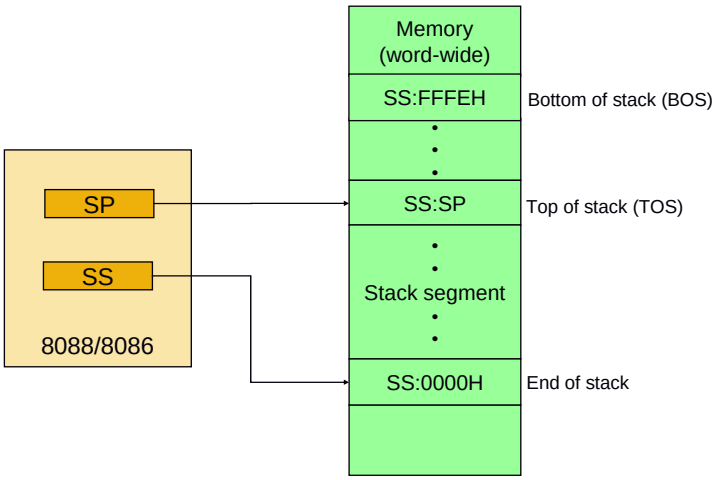
國立台灣大學
生物機電系
林達德

2.12 The Stack

- The ***stack*** is implemented for temporary storage of information such as data or addresses.
- The stack is 64KBytes long and is organized from a software point of view as 32K words.
- The contents of the SP and BP registers are used as offsets into the stack segment memory while the segment base value is in the SS register.
- Push instructions (PUSH) and pop instructions (POP)
- ***Top of the stack*** (TOS) and ***bottom of the stack*** (BOS)
- The 8088 can push **word-wide data** and address information onto the stack from registers or memory.
- Many stacks can exist but only one is active at a time.

國立台灣大學
生物機電系
林達德

2.12 The Stack



Stack segment of memory

國立台灣大學
生物機電系
林達德

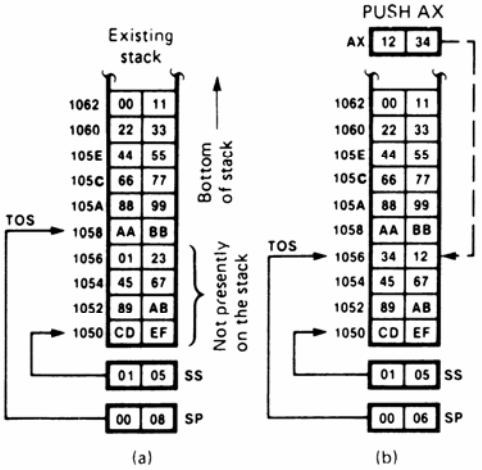
2.12 The Stack

EXAMPLE

- Push operation

$A_{BOS} = 01050_{16} + FFFE_{16}$
$= 01104_{16}$

$A_{TOS} = 01050_{16} + 0008_{16}$
$= 01058_{16}$

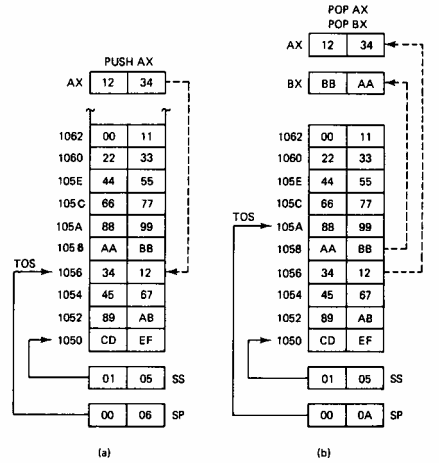


國立台灣大學
生物機電系
林達德

2.12 The Stack

EXAMPLE

- Pop operation



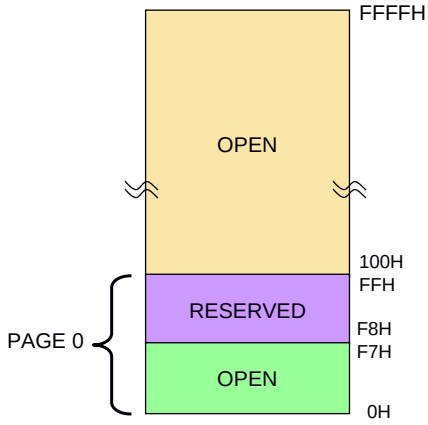
國立台灣大學
生物機電系
林達德

2.13 Input/Output Address Space

- The 8088 has separate memory and input/output (I/O) address space.
- The I/O address space is the place where I/O interfaces, such as printer and terminal ports, are implemented.
- The I/O address range is from 0000_{16} to $FFFF_{16}$. This represents 64KByte addresses.
- The I/O addresses are 16 bits long. Each of these addresses corresponds to one byte-wide I/O port.
- Certain I/O instructions can only perform operations to addresses 0000_{16} thru $00FF_{16}$ (***page 0***).

國立台灣大學
生物機電系
林達德

2.13 Input/Output Address Space



I/O address space

國立台灣大學
生物機電系
林達德