Computer Structure and Language

The 8086/8088 Assembly Language

Hamid Sarbazi-Azad
Department of Computer Engineering
Sharif University of Technology (SUT)
Tehran, Iran

8086/88 has 7 types of instructions:

1. Data Transfer Instructions
2. Arithmetic Instructions
3. Bit Manipulation Instructions
4. String Instructions
5. Program Execution Transfer Instructions
6. Processor Control Instructions
7. Interrupt Instructions

The 8086/88's String Processing Instructions:

OF	DF	IF	TF	SF	ZF	AF	PF	CF
-	-	-	-	-	-	-	-	-

- **Load String Byte/Word to AL/AX: LODS**

  Possible Use Cases: LODS label, LODSB, LODSW

  1010110 w

  AL ← $(M_{(SI)})$; if DF=1 then SI ← (SI)-1 else SI ← (SI)+1; if w=0
  AX ← $(M_{(SI)})$; if DF=1 then SI ← (SI)-2 else SI ← (SI)+2; if w=1

OF	DF	IF	TF	SF	ZF	AF	PF	CF
-	-	-	-	-	-	-	-	-

- **Store String Byte/Word from AL/AX: STOS**

  Possible Use Cases: STOS label, STOSB, STOSW

  1010101 w

  $M_{(DI)}$ ← (AL); if DF=1 then DI ← (DI)-1 else DI ← (DI)+1; if w=0
  $M_{(DI)}$ ← (AX); if DF=1 then DI ← (DI)-2 else DI ← (DI)+2; if w=1

OF	DF	IF	TF	SF	ZF	AF	PF	CF
X	-	-	-	X	X	X	X	X

- **Scan String Byte/Word: SCAS**

  Possible Use Cases: SCAS label, SCASB, SCASW

  1010111 w

  Realize (AL) - $(M_{(DI)})$ and update Flags; if DF=1 then DI ← (DI)-1 else DI ← (DI)+1; if w=0
  Realize (AX) - $(M_{(DI)})$ and update Flags; if DF=1 then DI ← (DI)-2 else DI ← (DI)+2; if w=1

* In all string instructions, the segment register used with SI is DS and segment register used with DI is ES.

The 8086/88's String Processing Instructions:

OF	DF	IF	TF	SF	ZF	AF	PF	CF
-	-	-	-	-	-	-	-	-

- **Move String Byte/Word: MOVS**

  Possible Use Cases: MOVS labels, MOVSB, MOVSW

  1010010 w

  $M_{(DI)}$ ← $(M_{(SI)})_{byte}$; if DF=1 then {DI ← (DI) -1; SI ← (SI)-1} if w=0
  else {DI ← (DI)+1; SI ← (SI) +1}

  $M_{(DI)}$ ← $(M_{(SI)})_{word}$; if DF=1 then {DI ← (DI) -2; SI ← (SI)-2} if w=1
  else {DI ← (DI)+2; SI ← (SI) +2}

OF	DF	IF	TF	SF	ZF	AF	PF	CF
X	-	-	-	X	X	X	X	X

- **Compare String Byte/Word: CMPS**

  Possible Use Cases: CMPS labels, CMPSB, CMPSW

  1010011 w

  Realize $(M_{(SI)})_{byte}$ - $(M_{(DI)})_{byte}$ and update Flags; if w=0
  if DF=1 then {DI ← (DI) -1; SI ← (SI)-1} else {DI ← (DI)+1; SI ← (SI) +1} }

  Realize $(M_{(SI)})_{word}$ - $(M_{(DI)})_{word}$ and update Flags; if w=1
  if DF=1 then {DI ← (DI) -2; SI ← (SI)-2} else {DI ← (DI)+2; SI ← (SI) +2}

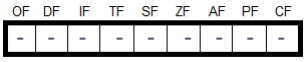
* In all string instructions, the segment register used with SI is DS and segment register used with DI is ES.

The 8086/88's String Processing Instructions:

- **Repeat Prefix: REP**
  Possible Use Cases: REP MOVS, REP LODS, REP STOS

  11110011

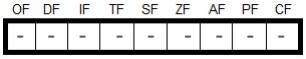
  1. If (CX)=0 then exit;
  2. Perform string instruction;
  3. CX ← (CX)-1; Goto 1;



- **Repeat on Equal/Zero Prefix: REPE/REPZ**
  Possible Use Cases: REPZ/REPE SCAS, REPZ/REPE CMPS

  11110011

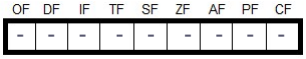
  1. If (CX)=0 or ZF=0 then exit;
  2. Perform string instruction;
  3. CX ← (CX)-1; Goto 1;



- **Repeat on Not Equal/Not Zero Prefix: REPNE/REPNZ**
  Possible Use Cases: REPNZ/REPNE SCAS, REPNZ/REPNE CMPS

  11110010

  1. If (CX)=0 or ZF=1 then exit;
  2. Perform string instruction;
  3. CX ← (CX)-1; Goto 1;



The 8086/88's Decimal Computing Instructions:

Decimal numbers can be shown in two formats:
- Packed Decimal: Each byte shows two decimal digits
- ASCII (unpacked) Decimal: Each byte shows one decimal digit

It is the programmers duty to keep the sign and define decimal numbers using **db** directive.

Machine has few limited instructions to support decimal calculations. They include AAA, AAS, AAM, AAD, DAA, DAS.

All instructions work on one byte decimal numbers
- 2 digits for packed decimal, and
- one digit for ASCII numbers).

The 8086/88's Decimal Computing Instructions:

ASCII (unpacked) Decimal Computing Instructions:

- **ASCII Adjust for Addition: AAA**

OF	DF	IF	TF	SF	ZF	AF	PF	CF
u	-	-	-	u	u	x	u	x

00110111

if (AL) ∧ 0Fh > 9 or AF=1 then AL ← (AL) + 6; AH ← (AH) + 1;
AF ← 1; CF ← AF;
AL ← (AL) ∧ 0Fh;

*AAA is used after actual binary addition.

- **ASCII Adjust for Subtraction: AAS**

OF	DF	IF	TF	SF	ZF	AF	PF	CF
u	-	-	-	u	u	x	u	x

00111111

if (AL) ∧ 0Fh > 9 or AF=1 then AL ← (AL) - 6; AH ← (AH) - 1;
AF ← 1; CF ← AF;
AL ← (AL) ∧ 0Fh;

*AAS is used after actual binary subtraction.

The 8086/88's Decimal Computing Instructions:

ASCII (unpacked) Decimal Computing Instructions:

- **ASCII Adjust for Muliplication: AAM**

OF	DF	IF	TF	SF	ZF	AF	PF	CF
u	-	-	-	x	x	u	x	u

11010100 00001010

AH ← (AL)/10; AL ← Remainder;

*AAM is used after actual binary multiplication.

- **ASCII Adjust for Division: AAD**

OF	DF	IF	TF	SF	ZF	AF	PF	CF
u	-	-	-	x	x	u	x	u

11010101 00001010

AL ← (AH)*10 + (AL); AH ← 0;

*AAD is used before actual binary division.

The 8086/88's Decimal Computing Instructions:

BCD (packed) Decimal Computing Instructions:

- **Decimal Adjust for Addition: DAA**

OF	DF	IF	TF	SF	ZF	AF	PF	CF
u	-	-	-	x	x	x	x	x

00100111

if (AL) ∧ 0Fh > 9 or AF=1 then AL ← (AL) + 6; AF ←1;
if (AL) > 9Fh or CF=1 then AL ← (AL) + 60h; CF ←1;

*DAA is used after actual binary Addition.

- **Decimal Adjust for Subtraction: DAS**

OF	DF	IF	TF	SF	ZF	AF	PF	CF
u	-	-	-	x	x	x	x	x

00101111

if (AL) ∧ 0Fh > 9 or AF=1 then AL ← (AL) - 6; AF ←1;
if (AL) > 9Fh or CF=1 then AL ← (AL) - 60h; CF ←1;

*DAS is used after actual binary Subtraction.

End of Slides