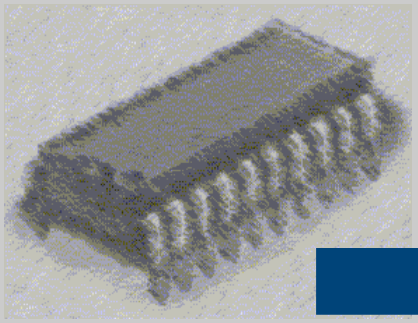
*Trabajo evaluable 1*

# Comparativas del número de ciclos de ejecución de un programa en ensamblador

***Arquitectura de Computadores***



I. T. Informática de Gestión

Curso 2009-2010

# Trabajo evaluable 1

## *Opción 1*

Sea el fragmento de programa escrito en ensamblador del i8086 siguiente:

```
CLD
LEA     SI, BUFFER1
LEA     DI, BUFFER2
MOV     CX,10
REP     MOVS    BUFFER2, BUFFER1
```

Dicho fragmento copia la cadena `buffer1` en la cadena `buffer2`, ambas de la misma longitud.

**Se pide:**

- Completar el fragmento anterior con las definiciones necesrias en el `.data` y completar el programa en el `.code`
- Realizar el mismo programa pero sin emplear instrucciones de manejo de cadenas
- Calcular para ambos programas el número de ciclos de instrucción necesarios para completarlos (consultar el apéndice en este mismo documento)
- Indicación de tiempos que se ha tardado en realizar el trabajo. Se deben indicar los tiempos (tanto de estudio en casa, en la biblioteca, consultas en internet, consultas en tutorías) empleados en: encontrar la solución, en redactar el trabajo y en crear la memoria.

## *Opción 2*

Sea el fragmento de programa escrito en ensamblador del i8086 siguiente:

```
MOV     AX, DS
MOV     ES, AX
CLD
MOV     AL, 'A'
MOV     CX,100
LEA     DI, Buffer1
REPNE   SCAS Buffer1
JE      Encontrado
NoEncontrado:     .
                  .
                  .
Encontrado:  DEC     DI
                  .
                  .
                  .
```

Dicho fragmento busca el carácter `A` en la cadena `buffer1` .

**Se pide:**

- Completar el fragmento anterior con las definiciones necesrias en el `.data` y completar el programa en el `.code`
- Realizar el mismo programa pero sin emplear instrucciones de manejo de cadenas
- Calcular para ambos programas el número de ciclos de instrucción necesarios para completarlos (consultar el apéndice en este mismo documento)
- Indicación de tiempos que se ha tardado en realizar el trabajo. Se deben indicar los tiempos (tanto de estudio en casa, en la biblioteca, consultas en internet, consultas en tutorías) empleados en: encontrar la solución, en redactar el trabajo y en crear la memoria.

# Apéndice sobre el cálculo de ciclos de las instrucciones

| Cálculo de la dirección efectiva en función del modo de direccionamiento | Ciclos en el i8088 | Ejemplo |
|---|---|---|
| Directo | 6 | MOV AX,ADDR |
| Relativo a registro<br>BX, BP, SI, DI | 5 | MOV AX,[BP] |
| Desplazamiento más registro base o registro índice<br>BX + Disp, BP + Disp<br>SI + Disp, DI + Disp | 9 | MOV AX,ADDR[BP] |
| Registro base + índice | | |
| BP + DI, BX + SI | 7 | MOV AX,[BP+DI] |
| BP + SI, BX + DI | 8 | MOV AX,[BX+DI] |
| Desplazamiento más regustri base + registro índice | | |
| BP + DI + Disp<br>BX + SI + Disp | 11 | MOV AX,ADDR[BP+DI] |
| BP + SI + Disp<br>BX + DI + Disp | 12 | MOV AX,ADDR[BP+SI] |

**Notas:**

- Añadir 2 ciclos de reloj si se sobreescriben segmentos
- Cada referencia a memoria requiere 4 ciclos adicionales

### Ciclos empleados por las instrucciones más usadas y de cadena.

Dónde pone EA se refiere a los cálculos necesarios para calcular los ciclos de la tabla superior

ADD

| Operands | Clocks Byte(word) | Transfers | Bytes | Example |
|---|---|---|---|---|
| register, register | 3 | - | 2 | ADD BX,CX |
| accumulator, immediate | 4 | - | 2-3 | ADD AX,256 |
| register, immediate | 4 | - | 3-4 | ADD BL,4 |
| register, memory | 9(13) + EA | 1 | 2-4 | ADD DI,[DX] |
| memory, register | 16(24) + EA | 2 | 2-4 | ADD TOTAL,BL |
| memory, immediate | 17(25) + EA | 2 | 3-6 | ADD RESULT,3 |

SUB

| Operands | Clocks byte(word) | Transfers | Bytes | Example |
|---|---|---|---|---|
| register, register | 3 | - | 2 | SUB DX,BX |
| register, memory | 9(13) + EA | 1 | 2-4 | SUB DX,TOTAL |
| memory, register | 16(24) + EA | 2 | 2-4 | SUB RATE,CL |
| accumulator, immediate | 4 | - | 2-3 | SUB AH,25 |
| register, immediate | 4 | - | 3-4 | SUB DX,5280 |
| memory, immediate | 17(25) + EA | 2 | 3-6 | SUB RES,1032 |

MUL

| Operands | Clocks | Transfers | Bytes | Example |
|---|---|---|---|---|
| reg8 | 70-77 | - | 2 | MUL CH |
| reg16 | 118-133 | - | 2 | MUL BX |
| mem8 | (76-83) + EA | 1 | 2-4 | MUL A_BYTE |
| mem16 | (128-143) + EA | 1 | 2-4 | MUL A_WORD |

IMUL

| Operands | Clocks | Transfers | Bytes | Example |
|---|---|---|---|---|
| reg8 | 80-98 | - | 2 | IMUL CL |
| reg16 | 128-154 | - | 2 | IMUL BX |
| mem8 | (86-104) + EA | 1 | 2-4 | IMUL BITE |
| mem16 | (138-164) + EA | 1 | 2-4 | IMUL WORD[BP][DI] |

DIV

| Operands | Clocks | Transfers | Bytes | Example |
|---|---|---|---|---|
| reg8 | 0-90 | - | 2 | DIV BL |
| reg16 | 4-162 | - | 2 | DIV BX |
| mem8 | (86-96) + EA | 1 | 2-4 | DIV VYUP |
| mem16 | (154-172) + EA | 1 | 2-4 | DIV NCONQUER[SI] |

IDIV

| Operands | Clocks | Transfers | Bytes | Example |
|---|---|---|---|---|
| reg8 | 101-112 | - | 2 | IDIV CL |
| reg16 | 165-184 | - | 2 | IDIV DX |
| mem8 | (107-118) + EA | 1 | 2-4 | IDIV BYTE[SI] |
| mem16 | (175-194) + EA | 1 | 2-4 | IDIV [BX].WORD_ARRAY |

INC

| Operands | Clocks byte(word) | Transfers | Bytes | Example |
|---|---|---|---|---|
| reg16 | 2 | - | 1 | INC BX |
| reg8 | 3 | - | 2 | INC BL |
| memory | 15(23) + EA | 2 | 2-4 | INC THETA[BX] |

DEC

| Operands | Clocks | Transfers | Bytes | Example |
|---|---|---|---|---|
| reg16 | 2 | - | 1 | DEC BX |
| reg8 | 3 | - | 2 | DEC BL |
| memory | 15(23) + EA | 2 | 2-4 | DEC MATRIX[SI] |

MOV

| Operands | Clocks byte(word) | Transfers | Bytes | Example |
|---|---|---|---|---|
| register, register | 2 | - | 2 | MOV BX,CX |
| memory, accumulator | 10(14) | 1 | 3 | MOV MEM_DEST,AL |
| accumulator, memory | 10(14) | 1 | 3 | MOV AX,MEM_SOURCE |
| memory, register | 9(13) + EA | 1 | 2-4 | MOV MEM_DEST,BX |
| register, memory | 8(12) + EA | 1 | 2-4 | MOV BX,MEM_SOURCE |
| register, immediate | 4 | - | 2-3 | MOV BX,0F6CDh |
| memory, immediate | 10(14) + EA | 1 | 3-6 | MOV MASK,0F6CDh |
| seg-reg, reg16 | 2 | - | 2 | MOV DS,BX |
| seg-reg, mem16 | 8(12) + EA | 1 | 2-4 | MOV DS,SEGMENT_VAL |
| reg16, seg-reg | 2 | - | 2 | MOV BP,SS |
| memory, seg-reg | 9(13) + EA | 1 | 2-4 | MOV CODE_VAR,CS |

PUSH

| Operands | Clocks | Transfers | Bytes | Example |
|---|---|---|---|---|
| register | 15 | 1 | 1 | PUSH BX |
| seg-reg (CS illegal) | 14 | 1 | 1 | PUSH ES |
| memory | 24 + EA | 2 | 2-4 | PUSH PARAMETERS |

POP

| Operands | Clocks | Transfers | Bytes | Example |
|---|---|---|---|---|
| register | 12 | 1 | 1 | POP CX |
| seg-reg (CS illegal) | 12 | 1 | 1 | POP ES |
| memory | 25 + EA | 2 | 2-4 | POP VALUE |

LEA

| Operands | Clocks | Transfers | Bytes | Example |
|---|---|---|---|---|
| reg16, mem16 | 2 + EA | - | 2-4 | LEA BX,MEM_ADDR |

NEG

| Operands | Clocks byte(word) | Transfers | Bytes | Example |
|---|---|---|---|---|
| register | 3 | - | 2 | NEG DL |
| memory | 16(24) + EA | 2 | 2-4 | NEG COEFFICIENT |

JZ/JE

| Operands | Clocks | Transfers | Bytes | Example |
|---|---|---|---|---|
| short-label | 16 or 4 | - | 2 | JE ZERO |

JNZ/JNE

| Operands | Clocks | Transfers | Bytes | Example |
|---|---|---|---|---|
| short-label | 16 or 4 | - | 2 | JNE NOT_EQUAL |

JG

| Operands | Clocks | Transfers | Bytes | Example |
|---|---|---|---|---|
| short-label | 16 or 4 | - | 2 | JG GREATER |

JL

| Operands | Clocks | Transfers | Bytes | Example |
|---|---|---|---|---|
| short-label | 16 or 4 | - | 2 | JG GREATER |

JLE

| Operands | Clocks | Transfers | Bytes | Example |
|---|---|---|---|---|
| short-label | 16 or 4 | - | 2 | JG GREATER |

JMP

| Operands | Clocks | Transfers | Bytes | Example |
|---|---|---|---|---|
| short-label | 15 | - | 2 | JMP ROPE_NEAR |
| near-label | 15 | - | 3 | JMP SAME_SEGMENT |
| far-label | 15 | - | 5 | JMP FAR_LABEL |
| memptr16 | 18 + EA | - | 2-4 | JMP SAME_SEG |
| regptr16 | 11 | - | 2 | JMP BX |
| memptr32 | 24 + EA | - | 2-4 | JMP NEXT_SEG |

CLD/STD

| Operands | Clocks | Transfers | Bytes | Example |
|---|---|---|---|---|
| no operands | 2 | - | 1 | CLD |

LOOP

| Operands | Clocks | Transfers | Bytes | Example |
|---|---|---|---|---|
| short-label | 17/5 | - | 2 | LOOP AGAIN |

LOOPE / LOOPZ

| Operands | Clocks | Transfers | Bytes | Example |
|---|---|---|---|---|
| short-label | 18 or 6 | - | 2 | LOOPE AGAIN |

LOOPNE / LOOPNZ

| Operands | Clocks | Transfers | Bytes | Example |
|---|---|---|---|---|
| short-label | 19 or 5 | - | 2 | LOOPNE AGAIN |

OR

| Operands | Clocks byte(word) | Transfers | Bytes | Example |
|---|---|---|---|---|
| register, register | 3 | - | 2 | OR CH,DL |
| register, memory | 9(13) + EA | 1 | 2-4 | OR BX, MEM_OR_Y |
| memory, register | 16(24) + EA | 2 | 2-4 | OR MEM_OR_Y, BX |
| accumulator, immediate | 4 | - | 2-3 | OR AL,01110110b |
| register,immediate | 4 | - | 3-4 | OR CX,00FFh |
| memory, immediate | 17(25) + EA | 2 | 3-6 | OR MEM_WORD,76h |

AND

| Operands | Clocks byte(word) | Transfers | Bytes | Example |
|---|---|---|---|---|
| register, register | 3 | - | 2 | AND AL,DL |
| register, immediate | 4 | - | 3-4 | AND CX,0FFh |
| accumulator, immediate | 4 | - | 2-3 | AND AX,01000010b |
| register, memory | 9(13) + EA | 1 | 2-4 | AND CX,MASK |
| memory, register | 16(24) + EA | 2 | 2-4 | AND VALUE,BL |
| memory, immediate | 17(25) + EA | 2 | 3-6 | AND STATUS,01h |

XOR

| Operands | Clocks | Transfers | Bytes | Example |
|---|---|---|---|---|
| register, register | 3 | - | 2 | XOR CX,BX |
| register, memory | 9(13) + EA | 1 | 2-4 | XOR CL,MASK_BYTE |
| memory, register | 16(24) + EA | 2 | 2-4 | XOR ALPHA[SI],DX |
| accumulator, immediate | 4 | - | 2-3 | XOR AL,01000001b |
| register, immediate | 4 | - | 3-4 | XOR SI,00C2h |
| memory, immediate | 17(25) + EA | 2 | 3-6 | XOR RETURN_CODE,0D2h |

NOT

| Operands | Clocks byte(word) | Transfers | Bytes | Example |
|---|---|---|---|---|
| register | 3 | - | 2 | NOT DX |
| memory | 16(24) + EA | 2 | 2-4 | NOT MASK |

SHL / SHR / SAL / SAR / ROL / ROR / RCL / RCR

| Operands | Clocks byte(word) | Transfers | Bytes | Example |
|---|---|---|---|---|
| register, 1 | 2 | - | 2 | SHL AL,1 |
| register, CL | 8 + 4/bit | - | 2 | SHL SI,CL |
| memory, 1 | 15(23) + EA | 2 | 2-4 | SHL WORD,1 |
| memory, CL | 20(28)+EA+4/bit | 2 | 2-4 | SHL BYTE,CL |

TEST

| Operands | Clocks byte(word) | Transfers | Bytes | Example |
|---|---|---|---|---|
| register, register | 3 | - | 2 | TEST SI,DX |
| register, memory | 9(13) + EA | 1 | 2-4 | TEST SI,MASK |
| accumulator, immediate | 4 | - | 2-3 | TEST AL,00000100b |
| register, immediate | 5 | - | 3-4 | TEST CX,1234 |
| memory, immediate | 11 + EA | - | 3-6 | TEST PARAM,1F1Fh |

CMP

| Operands | Clocks byte(word) | Transfers | Bytes | Example |
|---|---|---|---|---|
| register, register | 3 | - | 2 | CMP CX,BX |
| register, immediate | 4 | - | 3-4 | CMP BL,02h |
| accumulator, immediate | 4 | - | 2-3 | CMP AL,00010110b |
| register, memory | 9(13) + EA | 1 | 2-4 | CMP DH,ALPHA_BETA |
| memory, register | 9(13) + EA | 1 | 2-4 | CMP TOTAL,SI |
| memory, immediate | 10(14) + EA | 1 | 3-6 | CMP VALUES,3420h |

CALL

| Operands | Clocks byte(word) | Transfers | Bytes | Example |
|---|---|---|---|---|
| near-proc | 19(23) | 1 | 3 | CALL NEAR_PROC |
| far-proc | 28(36) | 2 | 5 | CALL FAR_PROC |
| memptr 16 | 21(29) + EA | 2 | 2-4 | CALL PROC_TABLE[SI] |
| regptr 16 | 16(24) | 1 | 2 | CALL AX |
| memptr 32 | 37(57) + EA | 4 | 2-4 | CALL [BX].ROUTINE |

RET

| Operands | Clocks | Transfers | Bytes | Example |
|---|---|---|---|---|
| (intrasegment, no pop) | 20 | 1 | 1 | RET |
| (intrasegment, with pop) | 24 | 1 | 3 | RET 4 |
| (intersegment, no pop) | 32 | 2 | 1 | RET |
| (intersegment, pop) | 31 | 2 | 3 | RET 2 |

INT

| Operands | Clocks byte(word) | Transfers | Bytes | Example |
|---|---|---|---|---|
| immed8 (type = 3) | 52 | 5 | 1 | INT 3 |
| immed8 (type <> 3) | 51 | 5 | 2 | INT 21 |

IRET

| Operands | Clocks | Transfers | Bytes | Example |
|---|---|---|---|---|
| no operands | 32 | 3 | 1 | IRET |

IN

| Operands | Clocks byte(word) | Transfers | Bytes | Example |
|---|---|---|---|---|
| accumulator, immed8 | 10(14) | 1 | 2 | IN AL,45h |
| accumulator, DX | 8(12) | 1 | 1 | IN AX,DX |

OUT

| Operands | Clocks byte(word) | Transfers | Bytes | Example |
|---|---|---|---|---|
| immed8, accumulator | 10(14) | 1 | 2 | OUT 254,AX |
| DX, accumulator | 8(12) | 1 | 1 | OUT DX,AL |

CMPS / CMPSB / CMPSW

| Operands | Clocks byte(word) | Transfers | Bytes | Example |
|---|---|---|---|---|
| dest,source | 22(30) | 2 | 1 | CMPS STR1,STR2 |
| (repeat) dest,source | 9 + 22(30)/rep | 2/rep | 1 | REPE CMPS STR1,STR2 |

LODS / LODSB / LODSW

| Operands | Clocks byte(word) | Transfers | Bytes | Example |
|---|---|---|---|---|
| source-str | 12(16) | - | 1 | LODS LIST |
| (repeat) source-str | 9+13(17)/rep | 1/rep | 1 | REP LODS LIST |

MOVS / MOVSB / MOVSW

| Operands | Clocks byte(word) | Transfers | Bytes | Example |
|---|---|---|---|---|
| dest,source | 18(26) | 2 | 1 | MOVS WORD_BUFF,INPUT |
| (repeat) dest,source | 9+17(25)/rep | 2/rep | 1 | REP MOVSW |

STOS / STOSB / STOSW

| Operands | Clocks byte(word) | Transfers | Bytes | Example |
|---|---|---|---|---|
| dest-string | 11(15) | 1 | 1 | STOS WORD_ARRAY |
| (repeat) dest-string | 9 + 10(14)/rep | 1/rep | 1 | REP STOS BYTE_ARRAY |

SCAS / SCASB / SCASW

| Operands | Clocks byte(word) | Transfers | Bytes | Example |
|---|---|---|---|---|
| dest-str | 15(19) | 1 | 1 | SCAS WORD_TABLE |
| (repeat) dest-str | 9 + 15(19)/rep | 1/rep | 1 | REPNE SCAS BYTE_TABLE |

REP / REPE / REPZ / REPNE / REPNZ

| Operands | Clocks | Transfers | Bytes | Example |
|---|---|---|---|---|
| - | 2 | - | 1 | REP MOVS TO,FROM |