Nombre y apellido: **Gerardo Tordoya**

DNI: **22.777.420**

* **Parte 1**: Consignas que implican la resolución de ejercicios con números de 4 bits y 8 bits.
* **Parte 2**: Preguntas relacionadas con los Trabajos Prácticos de la asignatura.

**CONSIGNAS**

**Primera parte**

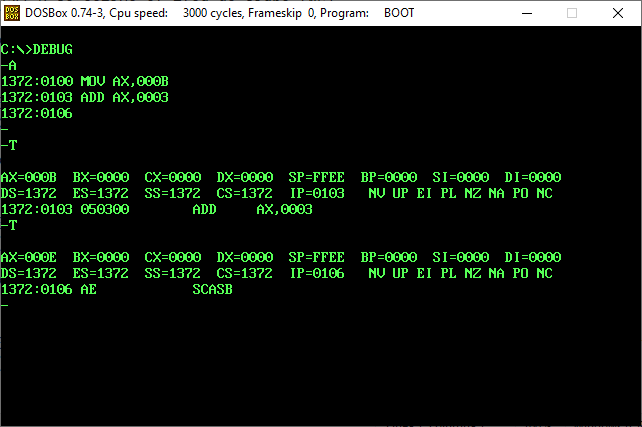
1. Suma binaria:

1.a) 1011 + 0011

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | **1** | **1** |  | DEC | HEX |
| VAL1 |  | 1 | 0 | 1 | 1 | 11 | B |
| VAL2 |  | 0 | 0 | 1 | 1 | 3 | 3 |
| RES |  | 1 | 1 | 1 | 0 | 14 | E |

1.b) ¿Cuáles fueron los flags que se activaron?

**Ninguno.** La bandera auxiliar se establece (en 1) si durante una operación de "agregar" hay un acarreo desde el nibble bajo (cuatro bits más bajos) al nibble alto (cuatro bits superiores). Como en esta operación no hay un nibble alto involucrado (la operación es de 4 bits), no hubo activación del Auxiliary Carry. Comprobación mediante el DEBUG:



1. Realice las siguientes operaciones por complemento a la base.

2.a) 0011 1001 – 0001 0001

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Valor a complementar |  | 0 | 0 | 0 | 1 |  | 0 | 0 | 0 | 1 |
| Complemento a 1 |  | 1 | 1 | 1 | 0 |  | 1 | 1 | 1 | 0 |
| Unidad |  |  |  |  |  |  |  |  |  | 1 |
| Complemento a 2 |  | 1 | 1 | 1 | 0 |  | 1 | 1 | 1 | 1 |
|  |  |  |  |  |  |  |  |  |  |  |
| Minuendo (sumando) |  | 0 | 0 | 1 | 1 |  | 1 | 0 | 0 | 1 |
| Sustraendo(sumando) |  | 1 | 1 | 1 | 0 |  | 1 | 1 | 1 | 1 |
| Valor final |  | 0 | 0 | 1 | 0 |  | 1 | 0 | 0 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |
| **Comprobación** |  |  |  |  |  |  |  |  |  |  |
| Minuendo a decimal |  | 0 | 0 | 1 | 1 |  | 1 | 0 | 0 | 1 |
| 57 |  |  |  | 32 | 16 |  | 8 |  |  | 1 |
| Sustraendo a decimal |  | 0 | 0 | 0 | 1 |  | 0 | 0 | 0 | 1 |
| 17 |  |  |  |  | 16 |  |  |  |  | 1 |
| Resta decimal |  |  |  |  |  |  |  |  |  |  |
| 40 |  |  |  |  |  |  |  |  |  |  |
| Obtenido a decimal |  | 0 | 0 | 1 | 0 |  | 1 | 0 | 0 | 0 |
| Conversión innecesaria |  | 0 | 0 | 1 | 0 |  | 1 | 0 | 0 | 0 |
| 40 |  |  |  | 32 |  |  | 8 |  |  |  |

n I / Pág. 3

2.b) 0000 0101 – 0001 1001

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Valor a complementar |  | 0 | 0 | 0 | 1 |  | 1 | 0 | 0 | 1 |
| Complemento a 1 |  | 1 | 1 | 1 | 0 |  | 0 | 1 | 1 | 0 |
| Unidad |  |  |  |  |  |  |  |  |  | 1 |
| Complemento a 2 |  | 1 | 1 | 1 | 0 |  | 0 | 1 | 1 | 1 |
|  |  |  |  |  |  |  |  |  |  |  |
| Minuendo (sumando) |  | 0 | 0 | 0 | 0 |  | 0 | 1 | 0 | 1 |
| Sustraendo (sumando) |  | 1 | 1 | 1 | 0 |  | 0 | 1 | 1 | 1 |
| Valor final |  | 1 | 1 | 1 | 0 |  | 1 | 1 | 0 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |
| **Comprobación** |  |  |  |  |  |  |  |  |  |  |
| Minuendo a decimal |  | 0 | 0 | 0 | 0 |  | 0 | 1 | 0 | 1 |
| 5 |  |  |  |  |  |  |  | 4 |  | 1 |
| Sustraendo a decimal |  | 0 | 0 | 0 | 1 |  | 1 | 0 | 0 | 1 |
| 25 |  |  |  |  | 16 |  | 8 |  |  | 1 |
| Resta decimal |  |  |  |  |  |  |  |  |  |  |
| -20 |  |  |  |  |  |  |  |  |  |  |
| Obtenido a decimal |  | 1 | 1 | 1 | 0 |  | 1 | 1 | 0 | 0 |
| Conversión directa |  | 0 | 0 | 0 | 1 |  | 0 | 1 | 0 | 0 |
| -20 |  |  |  |  | 16 |  |  | 4 |  |  |

1. Suponiendo que tiene que comparar dos números, determine con una operación aritmética el valor de los flags si el primer número es mayor, menor o igual que el segundo, y verifique la experimentación manual con el debug.

Citando al libro, para comparar dos números A y B, en la UAL se resta A-B. Suponiendo que son enteros, la indicación de la UAL permitirá conocer cómo es A respecto de B. Se explorarán dos casos (la resta de naturales y la de enteros) por recibir cada uno distinto tratamiento.

**NOTA:**

**Debe tenerse presente que la UAL en las restas invierte el CARRY.**

3.a) Experimentación manual

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Minuendo |  | 0 | 1 | 1 | 0 |  | DEC: | 06 | HEX: | 06 |
| Sustraendo |  | 0 | 0 | 1 | 1 |  | DEC: | 03 | HEX: | 03 |
|  |  |  |  |  |  |  |  |  |  |  |
| Minuendo |  | 0 | 1 | 1 | 0 |  | DEC: | 06 | HEX: | 06 |
| Complemento |  | 1 | 1 | 0 | 0 |  | DEC: | 12 | HEX: | C |
| Unidad |  |  |  |  | 1 |  | DEC: | 01 | HEX: | 01 |
| Total | **1** | 0 | 0 | 1 | 1 |  | DEC: | 03 | HEX: | 03 |
|  |  |  |  |  |  |  |  |  |  |  |
| **FLAGS** |  | **C** | **V** | **Z** | **S[[1]](#footnote-1)** |  | DEB: | 06 | - | 03 |
|  |  | **0** | **0** | **0** | **0** |  | PRN: |  |  | 03 |
|  |  |  |  |  |  |  |  |  |  |  |
| Naturales |  | 6 | - | 3 | = | 3 |  | M | ≥ | S |
| Enteros |  | 6 | - | 3 | = | 3 |  | M | ≥ | S |

**INDICADORES**

* Para Naturales: C=0 → M≥S
* Para Enteros: V=0 Y S=0 → M≥S
* Para Enteros: V=S y Z=0 → M≥S

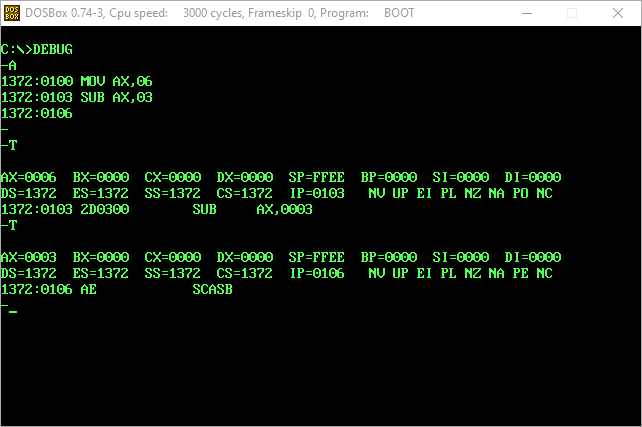
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Minuendo |  | 0 | 1 | 1 | 0 |  | DEC: | 06 | HEX: | 06 |
| Sustraendo |  | 0 | 1 | 1 | 0 |  | DEC: | 06 | HEX: | 06 |
|  |  |  |  |  |  |  |  |  |  |  |
| Minuendo |  | 0 | 1 | 1 | 0 |  | DEC: | 06 | HEX: | 06 |
| Complemento |  | 1 | 0 | 0 | 1 |  | DEC: | 09 | HEX: | 09 |
| Unidad |  |  |  |  | 1 |  | DEC: | 01 | HEX: | 01 |
| Total | **1** | 0 | 0 | 0 | 0 |  | DEC: | 00 | HEX: | 00 |
|  |  |  |  |  |  |  |  |  |  |  |
| **FLAGS** |  | **C** | **V** | **Z** | **S[[2]](#footnote-2)** |  | DEB: | 06 | - | 06 |
|  |  | **0** | **0** | **1** | **0** |  | PRN: |  |  | 00 |
|  |  |  |  |  |  |  |  |  |  |  |
| Naturales |  | 6 | - | 6 | = | 0 |  | M | = | S |
| Enteros |  | 6 | - | 6 | = | 0 |  | M | = | S |

**INDICADORES**

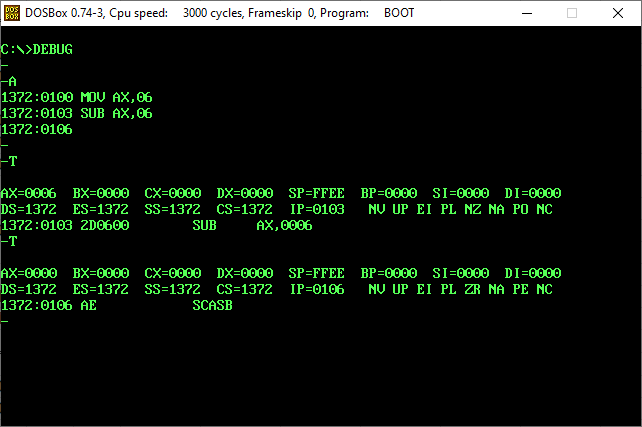
* Para Naturales: C=0 y Z=1 → M=S
* Para Enteros: V=0 y S=0 → M≥S
* Para Enteros: V=S y Z=1 → M=S

3.b) Experimentación con el debug

Pegue aquí las imágenes de la experimentación con el debug



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FLAGS** |  | **C** | **V** | **Z** | **S** |
|  | **0** | **0** | **0** | **0** |



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **FLAGS** |  | **C** | **V** | **Z** | **S** |
|  | **0** | **0** | **1** | **0** |

**Segunda parte**

En un lenguaje de alto nivel se tipeó:

**INTEGERS**

**R=P‐Q+T**

**Siendo P = +10 Q = +70 T = ‐30**

1. Los desplazamientos de las direcciones de memoria de las variables P, Q, T y R son respectivamente 2000, 3000, 4000 y 5000. El desplazamiento de la dirección de memoria de la primera instrucción a codificar es 0800. En función de los datos presentados codifique y escriba la zona de datos y la zona de instrucciones. Recuerde que los códigos de operación son los mismos que los que se utilizaron en los Trabajos Prácticos:

**A1** para transferir un dato de memoria al registro AX.

**0306** para sumarle al registro AX un número que está en una dirección de memoria

**2B06** para restarle al registro AX un número que está en una dirección de memoria

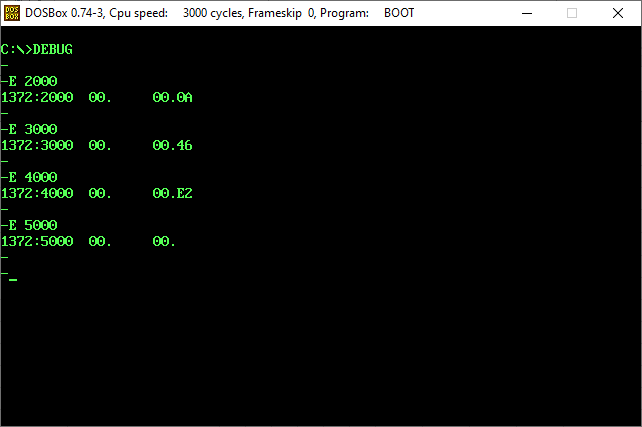
**A3** para pasar una copia del dato que está en el registro AX a memoria

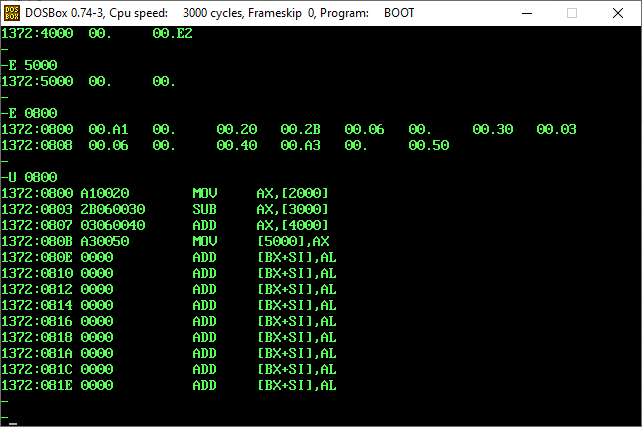
**Zonas de memoria de datos antes de la ejecución del programa**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dirección** | **Variable** | **Contenido** | | |
| **DEC** | **BIN** | **HEXA** |
| 2000 | P | 10 | 0000 | 00 |
| 2001 | 1010 | 0A |
| 3000 | Q | 70 | 0100 | 00 |
| 3001 | 0110 | 46 |
| 4000 | T | -30 | 1110 | 00 |
| 4001 | 0010 | E2 |

**Zonas de memoria de las instrucciones del programa.**

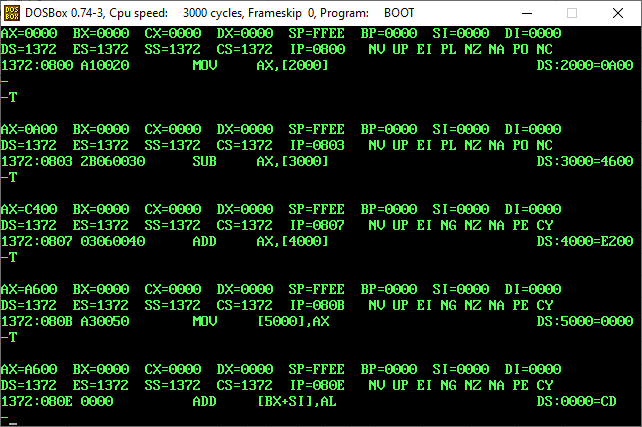
|  |  |
| --- | --- |
| **Dirección** | **Contenido** |
| 0800 | **A1** |
| 0801 | 00 |
| 0802 | 20 |
| 0803 | **2B** |
| 0804 | **06** |
| 0805 | 00 |
| 0806 | 30 |
| 0807 | **03** |
| 0808 | **06** |
| 0809 | 00 |
| 80A | 40 |
| 80B | **A3** |
| 80C | 00 |
| 80D | 50 |





1. Luego de ejecutar cada instrucción, indique el valor del registro AX en HEXA, el valor del

IP en HEXA y las FLAGS que se modificaron.

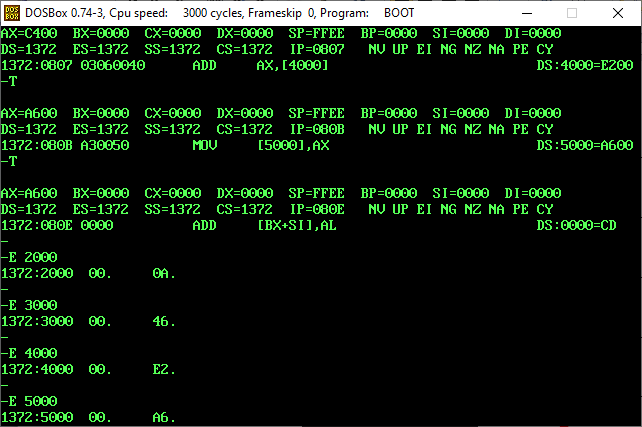


|  |  |  |  |
| --- | --- | --- | --- |
| **Instrucción** | **AX** | **IP** | **Flags** |
| I1 | 0A00 | 0803 | S=0 P=0 C=0 |
| I2 | C400 | 0807 | S=1 P=1 C=1 |
| I3 | A600 | 080B | S=1 P=1 C=1 |
| I4 | A600 | 080E | S=1 P=1 C=1 |

1. Indique cómo queda en memoria la zona de datos luego de ejecutar las instrucciones.

**Zonas de Datos de la Memoria**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dirección** | **Variable** | **Contenido** | | |
| **DEC** | **BIN** | **HEXA** |
| 2000 | P | 10 | 0000 | 00 |
| 2001 | 1010 | 0A |
| 3000 | Q | 70 | 0100 | 00 |
| 3001 | 0110 | 46 |
| 4000 | T | -30 | 1110 | 00 |
| 4001 | 0010 | E2 |
| 5000 | **R** | **-90** | **1010** | **00** |
| 5001 | **0110** | **A6** |



1. Suponiendo que se quiere imprimir el resultado en modo texto, indique cómo debe llegar a la placa de video lo que debe salir en pantalla, es decir R=xxxx

|  |  |
| --- | --- |
| **Carácter** | **ASCII** |
| R | 82 |
| = | 61 |
| - | 45 |
| 9 | 57 |
| 0 | 48 |

1. Signo (SF) queda en PL (positivo) cuando se establece en 0. [↑](#footnote-ref-1)
2. Signo (SF) queda en PL (positivo) cuando se establece en 0. [↑](#footnote-ref-2)