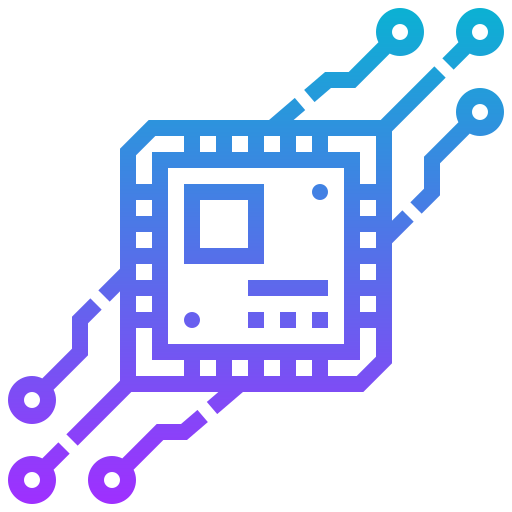
|  |
| --- |
| **TECNOLÓGICO DE ESTUDIOS SUPERIORES DE CHALCO**  Ingeniería en Sistemas Computacionales |



|  |  |
| --- | --- |
| **Lenguajes y autómatas 2** | **Salto Indexado** |
| **Arfaxad Zadot Aguilar Jiménez** | **Grupo 4701** |

Capítulo 1: Introducción a los autómatas

1.1 ¿Por qué estudiar la teoría de autómatas?

1.1.1 Introducción a los autómatas finitos

1.1.2 Representaciones estructurales

1.1.3 Autómatas y Complejidad

1.5 Conceptos fundamentales de la teoría de autómatas

1.5.1 Alfabetos

1.5.2 Cadenas de Caracteres

1.5.3 Lenguajes

1.5.4 Problemas

1.6 Resumen del Capítulo 1

1 INTRODUCCION A LOS AUTOMATAS

La teoría de autómatas es el estudio de dispositivos de cálculo abstractos, es decir, de las "máquinas”. En la década de

los años treinta, A. Turing estudió una máquina abstracta que tenía todas las capacidades de las computadoras de hoy día,

al menos en lo que respecta a lo que podían calcular. El objetivo de Turing era describir de forma precisa los límites entre

lo que una máquina de cálculo podía y no podía hacer; estas conclusiones no sólo se aplican a las máquinas abstractas de

Turing, sino a todas las máquinas reales actuales.

En las décadas de los años cuarenta y cincuenta, una serie de investigadores estudiaron las máquinas más simples, las

cuales todavía hoy denominamos "autómatas finitos". También a finales de la década de los cincuenta, el lingüista N.

Chomsky inició el estudio de las "gramáticas" formales. Aunque no son máquinas estrictamente, estas gramáticas están

estrechamente relacionadas con los autómatas abstractos y sirven actualmente como base de algunos importantes

componentes de software, entre los que se incluyen componentes de los compiladores

**INTRODUCCIÓN**

Capítulo 1: Introducción a los autómatas

1.1 ¿Por qué estudiar la teoría de autómatas?

1.1.1 Introducción a los autómatas finitos

1.1.2 Representaciones estructurales

1.1.3 Autómatas y Complejidad

1.5 Conceptos fundamentales de la teoría de autómatas

1.5.1 Alfabetos

1.5.2 Cadenas de Caracteres

1.5.3 Lenguajes

1.5.4 Problemas

1.6 Resumen del Capítulo 1

1 INTRODUCCION A LOS AUTOMATAS

La teoría de autómatas es el estudio de dispositivos de cálculo abstractos, es decir, de las "máquinas”. En la década de

los años treinta, A. Turing estudió una máquina abstracta que tenía todas las capacidades de las computadoras de hoy día,

al menos en lo que respecta a lo que podían calcular. El objetivo de Turing era describir de forma precisa los límites entre

lo que una máquina de cálculo podía y no podía hacer; estas conclusiones no sólo se aplican a las máquinas abstractas de

Turing, sino a todas las máquinas reales actuales.

En las décadas de los años cuarenta y cincuenta, una serie de investigadores estudiaron las máquinas más simples, las

cuales todavía hoy denominamos "autómatas finitos". También a finales de la década de los cincuenta, el lingüista N.

Chomsky inició el estudio de las "gramáticas" formales. Aunque no son máquinas estrictamente, estas gramáticas están

estrechamente relacionadas con los autómatas abstractos y sirven actualmente como base de algunos importantes

componentes de software, entre los que se incluyen componentes de los compiladores

Capítulo 1: Introducción a los autómatas

1.1 ¿Por qué estudiar la teoría de autómatas?

1.1.1 Introducción a los autómatas finitos

1.1.2 Representaciones estructurales

1.1.3 Autómatas y Complejidad

1.5 Conceptos fundamentales de la teoría de autómatas

1.5.1 Alfabetos

1.5.2 Cadenas de Caracteres

1.5.3 Lenguajes

1.5.4 Problemas

1.6 Resumen del Capítulo 1

1 INTRODUCCION A LOS AUTOMATAS

La teoría de autómatas es el estudio de dispositivos de cálculo abstractos, es decir, de las "máquinas”. En la década de

los años treinta, A. Turing estudió una máquina abstracta que tenía todas las capacidades de las computadoras de hoy día,

al menos en lo que respecta a lo que podían calcular. El objetivo de Turing era describir de forma precisa los límites entre

lo que una máquina de cálculo podía y no podía hacer; estas conclusiones no sólo se aplican a las máquinas abstractas de

Turing, sino a todas las máquinas reales actuales.

En las décadas de los años cuarenta y cincuenta, una serie de investigadores estudiaron las máquinas más simples, las

cuales todavía hoy denominamos "autómatas finitos". También a finales de la década de los cincuenta, el lingüista N.

Chomsky inició el estudio de las "gramáticas" formales. Aunque no son máquinas estrictamente, estas gramáticas están

estrechamente relacionadas con los autómatas abstractos y sirven actualmente como base de algunos importantes

componentes de software, entre los que se incluyen componentes de los compiladores

El universo de los sistemas automatizados, digital, virtual. Cada día absorbe más terreno y forma parte ya de nosotros. Por ejemplo, en el interior de nuestros hogares, encontramos lavadoras, hornos microondas equipos de sonido, video, sistemas de calefacción/aire acondicionado etc.

**OBJETIVO**

Diseñar el código para la siguiente tabla de verdad. Para cada cantidad de votos.

* A = 10 %
* B = 20%
* C = 30%
* D = 40%

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| D | C | B | A | S |
| 0 | 0 | **0** | **0** | 0 |
| 0 | 0 | **0** | **1** | 0 |
| 0 | 0 | **1** | **0** | 0 |
| 0 | 0 | **1** | **1** | 0 |
| 0 | 1 | **0** | **0** | 0 |
| 0 | 1 | **0** | **1** | 0 |
| 0 | 1 | **1** | **0** | 0 |
| 0 | 1 | **1** | **1** | 1 |
| 1 | 0 | **0** | **0** | 0 |
| 1 | 0 | **0** | **1** | 0 |
| 1 | 0 | **1** | **0** | 1 |
| 1 | 0 | **1** | **1** | 1 |
| 1 | 1 | **0** | **0** | 1 |
| 1 | 1 | **0** | **1** | 1 |
| 1 | 1 | **1** | **0** | 1 |
| 1 | 1 | **1** | **1** | 1 |

**CÓDIGO**

\_\_CONFIG \_CP\_OFF & \_WDT\_OFF & \_PWRTE\_ON & \_XT\_OSC

LIST P=16F84A

INCLUDE <P16F84A.INC>

;ZONA DE CODIGO

ORG 0X0

BSF STATUS,RP0

MOVLW B'00011111'

MOVWF TRISA

CLRF TRISB

BCF STATUS, RP0

PRINCIPAL

MOVF PORTA,W

ANDLW B'00001111'

CALL TABLA

MOVWF PORTB

GOTO PRINCIPAL

TABLA

ADDWF PCL,F

RETLW D'0'

RETLW D'0'

RETLW D'0'

RETLW D'0'

RETLW D'0'

RETLW D'0'

RETLW D'0'

RETLW D'1'

RETLW D'0'

RETLW D'0'

RETLW D'1'

RETLW D'1'

RETLW D'1'

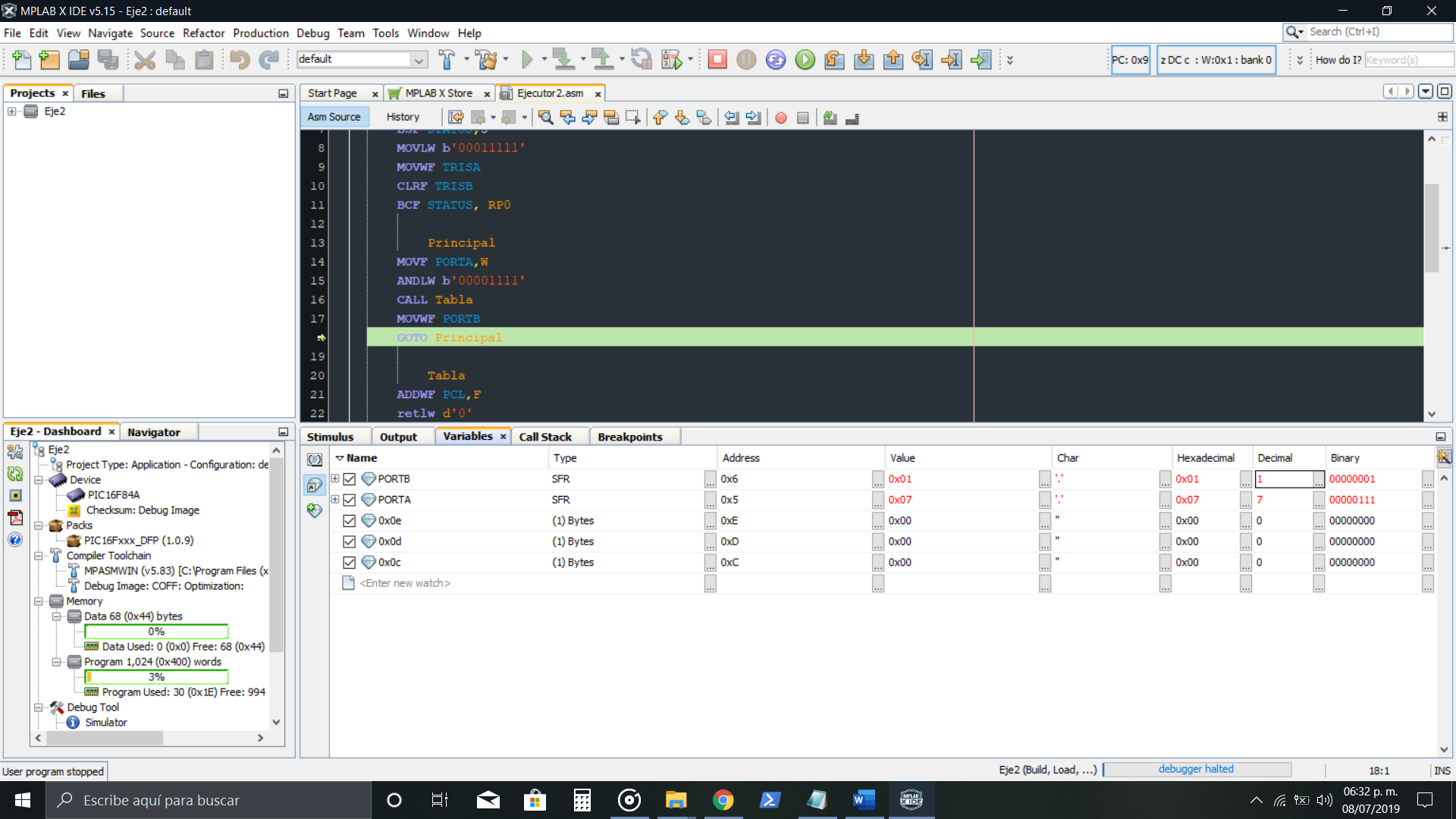
RETLW D'1'

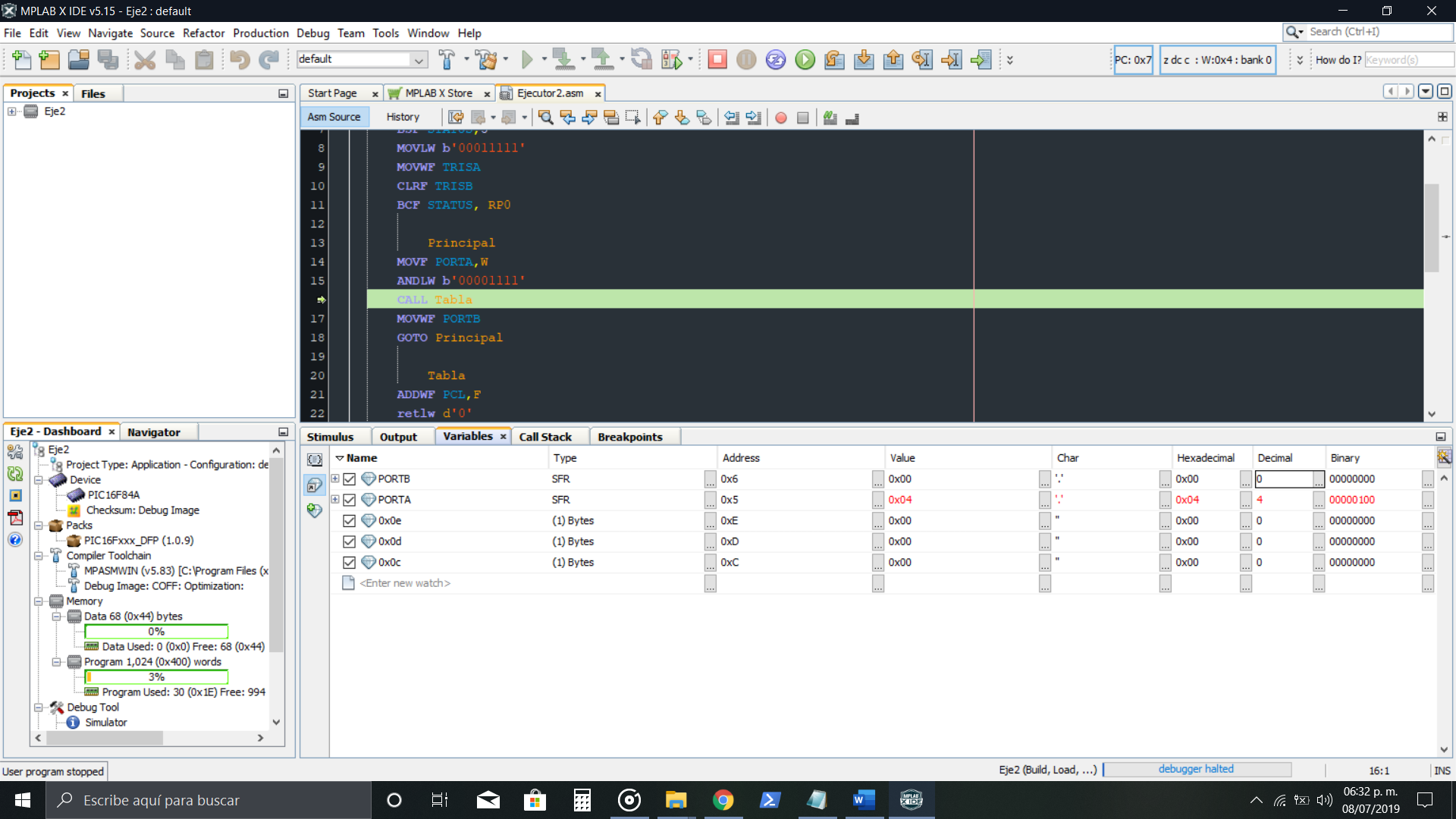
RETLW D'1'

RETLW D'1'

END

**SIMULACIÓN**





**CONCLUSIÓN**

El salto indexado llega a resultar muy útil si requiere que el programa prenda una cantidad de bits en especifico como fue el caso de la tabla de verdad ya que con operandos de suma y resta no son suficientes para cumplir la tarea.