

CENTRO REGIONAL UNIVERSITARIO CÓRDOBA IUA



“Cyclone IV”

ARQ. DE COMPUTADORAS I - INGENIERÍA EN INFORMÁTICA

PROYECTO INTEGRADOR

PROFESOR

Toledo, Luis Eduardo

INTEGRANTES

Carrizo, Matías (mcarrizo490@alumnos.iua.edu.ar)

Guillaumet, Gabriel (gguillaumet011@alumnos.iua.edu.ar)

14 de noviembre del año 2022



Índice General

Índice General	1
Introducción	2
Objetivos	3
Desarrollo	4
Observaciones	4
Funcionamiento	6
Instrucciones	10
Menú	10
Selección	10
Secuencias	12
Conclusiones	16
Bibliografía	17



Introducción

El siguiente trabajo consiste en la implementación de conceptos trabajados en la materia Arquitectura de Computadoras 1. La finalidad del proyecto es manipular una placa DE0-Nano con funcionalidades que permitan la interacción del usuario con secuencias de luces. Toda la codificación del proyecto se encuentra dada en el lenguaje de programación *Assembly*.



Objetivos

Este proyecto tiene como finalidad los siguientes objetivos:

- Realizar un menú, el cual dé a elegir cuatro secuencias de luces diferentes, dos de ellas tienen que ser el “Auto fantástico”, ejecutada a través de un algoritmo, y el “Choque” ejecutada mediante tabla. Las otras dos, deberán crearlas los alumnos teniendo el mismo formato que las anteriores.
- Implementar el menú y las cuatro secuencias en el simulador del IDE Quartus 1.19.
- Cargar el código en hexadecimal a la placa “De0-nano” y ejecutarlo.
- Realizar una opción para cambiar las velocidades de las secuencias.

Desarrollo

A continuación vamos a explicar la operatoria que seguimos para llevar a cabo el proyecto

Observaciones

➤ Distribución de pines

- INport: A continuación se detalla los pines correspondientes a la señal de entrada *INport* del módulo *top*

1	1	1	E1	M15	B9	T8	M1
---	---	---	----	-----	----	----	----

Observar que, los tres bits más significativos no han sido asignados a ningún pin, lo que implica que, por defecto, su valor será 1.

- OUTport: A continuación se detalla los pines correspondientes a la señal de salida *OUTport* del módulo *top*

L3	B1	F3	D1	A11	B13	A13	A15
----	----	----	----	-----	-----	-----	-----

- resetE: Para esta señal utilizamos el botón key_0 asignado al pin J15.
- clk: Para esta señal utilizamos el pin R8 asignado al clock de 50 MHz propio de la placa.

➤ Funcionamiento de la placa

Dado que los pines E1 y J15, correspondientes a los botones key_0 y key_1 tienen funcionamiento por bajo (es decir, mientras no están presionados, la señal que emiten corresponde a un uno lógico, mientras que cuando son presionados, emiten un cero lógico) realizamos la siguiente corrección al código del ARM reducido implementado para llevar a cabo el proyecto.

```

6 - module top(input logic clk, reset,
7             input logic [7:0] INport,
8             output logic [7:0] OUTport);
9
10 logic [31:0] WriteData, DataAdr;
11 logic MemWrite, MemtoReg, PortSel;
12 logic [31:0] PC, Instr, ReadData, MemData;
13 logic [7:0] INData;
14

```

```

6 + module top(input logic clk, resetE,
7             input logic [7:0] INport,
8             output logic [7:0] OUTport);
9
10 logic [31:0] WriteData, DataAdr;
11 logic MemWrite, MemtoReg, PortSel;
12 logic [31:0] PC, Instr, ReadData, MemData;
13 logic [7:0] INData;
14
15 + logic reset;
16 + assign reset = ~resetE;

```

La corrección consiste en la negación de la señal externa del clock (resetE en la imagen de la derecha) proveniente de la placa, valor que luego asignamos a una señal interna (reset) para sustituir la señal original.

➤ Lectura de tabla de datos

Para poder implementar secuencias leídas desde tabla, tuvimos que insertar las siguientes líneas de código al ARM reducido.

```
module dmem(input logic clk, we,
            input logic [31:0] a, wd,
            output logic [31:0] rd);

logic [31:0] RAM[63:0];

initial
    $readmemh("E:\\Desktop\\Arquitectura\\dmem.dat", RAM);

assign rd = RAM[a[31:2]]; // word aligned

always_ff @(posedge clk)
    if (we) RAM[a[31:2]] <= wd;
endmodule
```

permitiendo así la lectura de una tabla de datos.

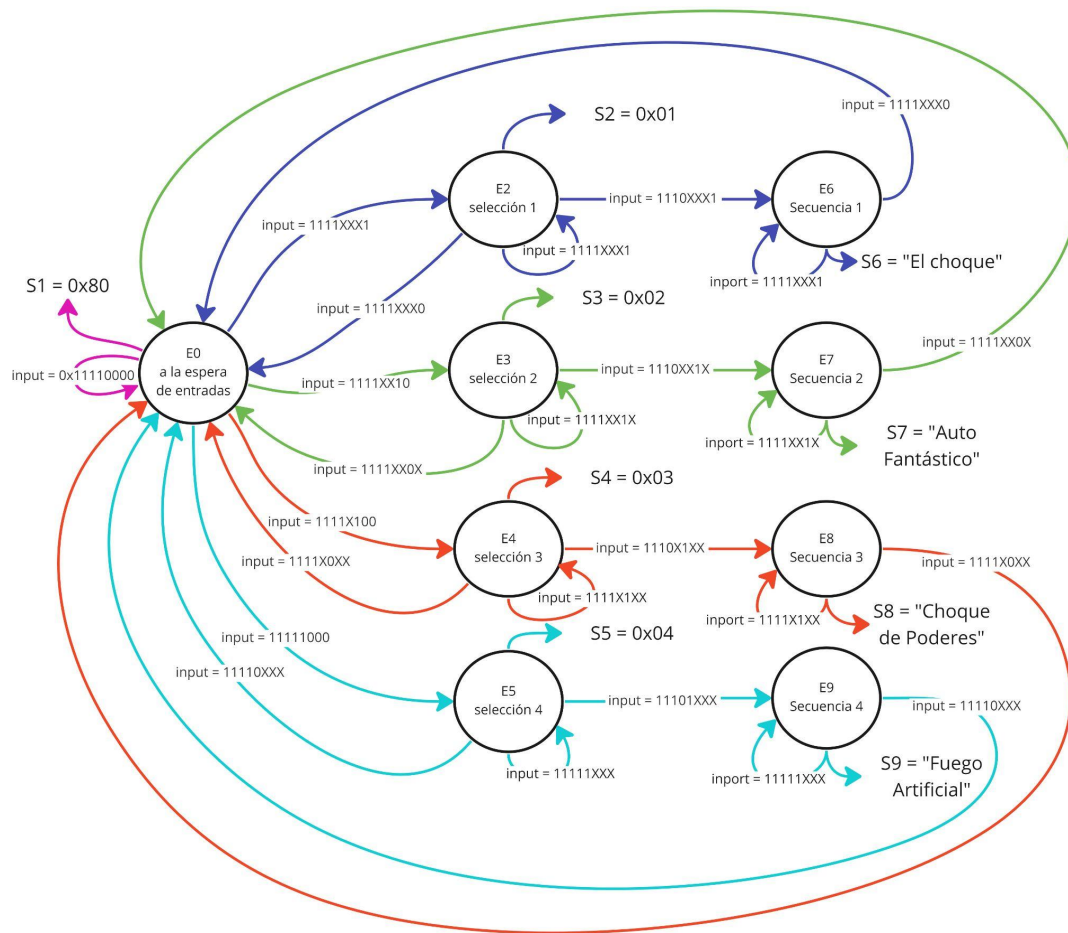
Funcionamiento

- Entradas

Mediante la siguiente tabla presentamos las distintas combinaciones de la señal *INport* y su correspondencia a las distintas funcionalidades del programa.

Operación	Pin E1	Pin M15	Pin B9	Pin T8	Pin M1
Selección 1	1	X	X	X	1
Salir Sel. 1	1	X	X	X	0
Selección 2	1	X	X	1	0
Salir Sel. 2	1	X	X	0	X
Selección 3	1	X	1	0	0
Salir Sel. 3	1	X	0	X	X
Selección 4	1	1	0	0	0
Salir Sel. 4	1	0	X	X	X
Secuencia 1	0	X	X	X	1
Salir Sec. 1	1	X	X	X	0
Secuencia 2	0	X	X	1	X
Salir Sec. 2	1	X	X	0	X
Secuencia 3	0	X	1	X	X
Salir Sec. 3	1	X	0	X	X
Secuencia 4	0	1	X	X	X
Salir Sec. 4	0	0	X	X	X

Esta tabla también puede ser representada mediante el siguiente diagrama de estados



- Ilustración de secuencias



```

1  dmem.dat
2
3  00000000 --> Sin usar
4  003f0000 --> Valor del delay

6  /* Secuencia: El Choque */
7
8  Con valores de tabla
9
10 00000081 --> * - - - - - *
11 00000042 --> - * - - - - * -
12 00000024 --> - - * - - * - -
13 00000018 --> - - - * * - - -
14 00000018 --> - - - * * - - -
15 00000024 --> - - * - - * - -
16 00000042 --> - * - - - - * -
17 00000081 --> * - - - - - *
18
19 /* Secuencia: Choque de Poderes */
20
21 Con valores de tabla
22
23 00000081 --> * - - - - - *      00000001 --> - - - - - - *
24 00000042 --> - * - - - - * -    00000003 --> - - - - - - * *
25 00000024 --> - - * - - * - -    00000007 --> - - - - - * * *
26 00000018 --> - - - * * - - -    0000000F --> - - - - * * * *
27 0000000C --> - - - - * * - -    0000001F --> - - - * * * * *
28 00000018 --> - - - * * - - -    0000003F --> - - * * * * * *
29 00000030 --> - - * * - - - -    0000007F --> - * * * * * * *
30 00000060 --> - * * - - - - -    000000FF --> * * * * * * * *
31 00000030 --> - - * * - - - -    000000FE --> * * * * * * * -
32 00000018 --> - - - * * - - -    000000FC --> * * * * * * - -
33 0000000C --> - - - - * * - -    000000F8 --> * * * * * - - -
34 00000006 --> - - - - - * * -    000000F0 --> * * * * - - - -
35 00000018 --> - - - * * - - -    000000E0 --> * * * - - - - -
36 00000006 --> - - - - - * * -    000000C0 --> * * - - - - - -
37 00000003 --> - - - - - * *      00000080 --> * - - - - - - -

```



```

39  /* Secuencia: Auto fantástico */
40
41  Por algoritmo
42
43  00000080 --> * - - - - - 00000002 --> - - - - - * -
44  00000040 --> - * - - - - 00000004 --> - - - - - * -
45  00000020 --> - - * - - - 00000008 --> - - - - - * -
46  00000010 --> - - - * - - 00000010 --> - - - * - -
47  00000008 --> - - - - * - 00000020 --> - - * - - -
48  00000004 --> - - - - - * - 00000040 --> - * - - - -
49  00000002 --> - - - - - * - 00000080 --> * - - - - -
50  00000001 --> - - - - - *
51
52  /* Secuencia: Fuego Artificial */
53
54  Por algoritmo
55
56  10000000 --> * - - - - -
57  01000000 --> - * - - - -
58  00100000 --> - - * - - -
59  00010000 --> - - - * - -
60  00001000 --> - - - - * -
61  00010100 --> - - - * - * -
62  00100010 --> - - * - - * -
63  01000001 --> - * - - - *
64  00000000 --> - - - - -

```



Instrucciones

Menú

```

1  /----- Menu -----/
2
3  inicio:  SUB    R0, R15, R15      ; E04F000F ; Utilizamos el PC para mover un 0 a R0
4          ADD    R2, R0, #0x80    ; E2802080 ; Movemos a R2 el valor #0x80, sera nuestra luz del menu
5          STR     R2, [R0, #0x800] ; E5802800 ; Sacamos al puerto 0x800, correspondiente a la placa, el registro R2
6
7  vuelta:  ADD    R3, R0, #2       ; E2803002 ; Movemos un 2 al registro R3, contador para el bucle de lectura
8
9  lectura: SUBS    R3, R3, #1       ; E2533001 ; Restamos uno al contador y actualizamos banderas
10         LDR     R1, [R0, #0x800] ; E5901800 ; Leemos el puerto, guardando en R1 los valores de los switches y del key_1
11         BNE     lectura          ; 1AFFFFFC ;
12
13         ANDS    R2, R1, #0x0F    ; E211200F ; Primer mascara, verificamos la presencia de switches encendidos
14         BEQ     vuelta          ; 0AFFFFF9 ;
15
16         ANDSNE  R2, R1, #0x01    ; 12112001 ; Segunda mascara, verificamos pin M1 encendido
17         STRNE   R0, [R0, #0x800] ; 15800800 ; Apagamos luces del menu
18         BNE     sel1            ; 1A000009 ;
19
20         ANDSEQ  R2, R1, #0x02    ; 02112002 ; Tercer mascara, verificamos pin T8 encendido
21         STRNE   R0, [R0, #0x800] ; 15800800 ; Apagamos luces del menu
22         BNE     sel2            ; 1A000011 ;
23
24         ANDSEQ  R2, R1, #0x04    ; 02112004 ; Cuarta mascara, verificamos pin B9 encendido
25         STRNE   R0, [R0, #0x800] ; 15800800 ; Apagamos luces del menu
26         BNE     sel3            ; 1A000019 ;
27
28         ANDSEQ  R2, R1, #0x08    ; 02112008 ; Quinta mascara, verificamos pin M15 encendido
29         STRNE   R0, [R0, #0x800] ; 15800800 ; Apagamos luces del menu
30         BNE     sel4            ; 1A000021 ;
31         B       inicio          ; EAFFFFEA ; Si ninguna condicion se cumplio, volvemos al inicio

```

Selección

```

35 /-----/
36 /----- Seleccion 1 -----/
37 /-----/
38
39 sel1:    AND     R2, R1, #0x01    ; E2012001 ; Movemos un 1 a R2
40         STR     R2, [R0, #0x800] ; E5802800 ; Sacamos al puerto la luz 1, correspondiente a la opcion elegida
41
42 vuelta:  ADD     R3, R0, #2       ; E2803002 ; Movemos un 2 al registro R3, contador para el bucle de lectura
43
44 lectura: SUBS    R3, R3, #1       ; E2533001 ; Restamos uno al contador y actualizamos banderas
45         LDR     R1, [R0, #0x800] ; E5901800 ; Leemos el puerto, guardando en R1 los valores de los switches y del key_1
46         BNE     lectura          ; 1AFFFFFC ;
47
48         ANDS    R2, R1, #0x10    ; E2112010 ; Primer mascara, verificamos pin E1 apagado (boton presionado)
49         BEQ     sec1            ; 0A000023 ;
50         ANDSNE  R2, R1, #0x01    ; 12112001 ; Segunda mascara, verificamos pin M1 apagado
51         BEQ     inicio          ; 0AFFFFE0 ;
52         BNE     vuelta          ; 1AFFFFF6 ; Si el pin M1 sigue encendido, esperamos a E1 apagado o M1 apagado
53

```



```

54 /-----/
55 /----- Seleccion 2 -----/
56 /-----/
57
58 sel2:      AND    R2, R1, #0x02      ; E2012002 ; Movemos un 2 a R2
59 |          STR    R2, [R0, #0x800]   ; E5802800 ; Sacamos al puerto la luz 2, correspondiente a la opcion elegida
60
61 vuelta:    ADD    R3, R0, #2         ; E2803002 ; Movemos un 2 al registro R3, contador para el bucle de lectura
62
63 lectura:   SUBS    R3, R3, #1         ; E2533001 ; Restamos uno al contador y actualizamos banderas
64 |          LDR     R1, [R0, #0x800]   ; E5901800 ; Leemos el puerto, guardando en R1 los valores de los switches y del key_1
65 |          BNE     lectura            ; 1AFFFFFFC ;
66
67           ANDS    R2, R1, #0x10      ; E2112010 ; Primer mascara, verificamos pin E1 apagado (boton presionado)
68           BEQ     sec2                ; 0A00002B ;
69           ANDSNE  R2, R1, #0x02      ; 12112002 ; Segunda mascara, verificamos pin T8 apagado
70           BEQ     inicio              ; 0AFFFFD5 ;
71           BNE     vuelta              ; 1AFFFFFF6 ; Si el pin M1 sigue encendido, esperamos a E1 apagado o M1 apagado
72
73 /-----/
74 /----- Seleccion 3 -----/
75 /-----/
76
77 sel3:      AND    R2, R1, #0x04      ; E2012004 ; Movemos un 4 a R2
78 |          STR    R2, [R0, #0x800]   ; E5802800 ; Sacamos al puerto la luz 3, correspondiente a la opcion elegida
79
80 vuelta:    ADD    R3, R0, #2         ; E2803002 ; Movemos un 2 al registro R3, contador para el bucle de lectura
81
82 lectura:   SUBS    R3, R3, #1         ; E2533001 ; Restamos uno al contador y actualizamos banderas
83 |          LDR     R1, [R0, #0x800]   ; E5901800 ; Leemos el puerto, guardando en R1 los valores de los switches y del key_1
84 |          BNE     lectura            ; 1AFFFFFFC ;
85
86           ANDS    R2, R1, #0x10      ; E2112010 ; Primer mascara, verificamos pin E1 apagado (boton presionado)
87           BEQ     sec3                ; 0A000048 ;
88
89           ANDSNE  R2, R1, #0x04      ; 12112004 ; Segunda mascara, verificamos pin B9 apagado
90           BEQ     inicio              ; 0AFFFFCA ;
91           BNE     vuelta              ; 1AFFFFFF6 ; Si el pin M1 sigue encendido, esperamos a E1 apagado o M1 apagado
92
93 /-----/
94 /----- Seleccion 4 -----/
95 /-----/
96
97 ~ sel4:     AND    R2, R1, #0x08      ; E2012008 ; Movemos un 8 a R2
98 |          STR    R2, [R0, #0x800]   ; E5802800 ; Sacamos al puerto la luz 4, correspondiente a la opcion elegida
99
100 vuelta:    ADD    R3, R0, #2         ; E2803002 ; Movemos un 2 al registro R3, contador para el bucle de lectura
101
102 ~ lectura:  SUBS    R3, R3, #1         ; E2533001 ; Restamos uno al contador y actualizamos banderas
103 |          LDR     R1, [R0, #0x800]   ; E5901800 ; Leemos el puerto, guardando en R1 los valores de los switches y del key_1
104 |          BNE     lectura            ; 1AFFFFFFC ;
105
106           ANDS    R2, R1, #0x10      ; E2112010 ; Primer mascara, verificamos pin E1 apagado (boton presionado)
107           BEQ     sec4                ; 0A00004F ;
108
109           ANDSNE  R2, R1, #0x08      ; 12112008 ; Segunda mascara, verificamos pin M15 apagado
110           BEQ     inicio              ; 0AFFFFBF ;
111           BNE     vuelta              ; 1AFFFFFF6 ; Si el pin M1 sigue encendido, esperamos a E1 apagado o M1 apagado
112

```



Secuencias

```

113 /-----/
114 /----- Secuencia 1 -----/
115 /-----/
116
117 sec1:      LDR r5, [r0, #4]          ; E5905004 ; Cargamos el valor de tabla del delay
118           ADD r7, r0, #12          ; E2807008 ; Guardamos el valor de la primer palabra de la secuencia
119           ADD r8,r0,#8             ; E2808008 ; Contador de palabras de la tabla
120
121 loop1:     LDR r9, [r7]              ; E5979000 ; Lee el dato guardado en R7 (led a prender)
122           STR r9, [r0, #0x800]      ; E5809800 ; Saca la luz al puerto
123           ADD r6,r0,r5              ; E0806005 ; Guardamos en r6 el valor del delay en r5
124
125 delay:     SUBS r6, r6, #1           ; E2566001 ; Restamos hasta llegar a cero
126           BNE #0x120               ; 1AFFFFFD ;
127
128           ADD r7, r7, #4            ; E2877004 ; Aumentamos en 4 para pasar a la proxima palabra de tabla
129           SUBS r8,r8,#1             ; E2588001 ; Restamos 1 al contador de instrucciones
130           ADDEQ r7,r0,#12           ; 02807008 ; Si llega a cero, suma la cantidad de lineas de la tabla
131           ADDEQ r8,r8,#8            ; 02888008 ; Y vuelve a la primer palabra de la secuencia
132           ADD R3,R0,#0x2            ; E2803002 ; Contador de doble lectura
133
134 lectura:   SUBS R3,R3,#1            ; E2533001 ; Lectura del puerto
135           LDR R1,[R0,#0x800]        ; E5901800 ;
136           BNE lectura              ; 1AFFFFFC ;
137
138           ANDS R1,R1,#0x01          ; E2111001 ; Mascara: verifica que se haya apagado el pin_M1
139           BEQ inicio                ; 0AFFFFAC ;
140           BNE loop1                ; 1AFFFFE7 ; Si no se apaga el PIN_M1 vuelve a la secuencia

```



```

142 /-----/
143 /----- Secuencia 2 -----/
144 /-----/
145
146 sec2:      ADD    R4, R0, #7      ; E2804007 ; Movemos un 7 al registro R4, contador del loop1
147           ADD    R8, R0, #0      ; E2808000 ; Movemos un 0 al registro R8, contador de luces encendidas
148           ADD    R6, R0, #0x80    ; E2806080 ; Movemos el valor 0x80 al registro R6, este sera nuestro registro de luces
149
150 loop1:     STR    R6, [R0, #0x800] ; E5806800 ; Sacamos la luz de la secuencia al puerto
151           ADD    R5, R0, #6      ; E2805006 ; Movemos un 6 al registro R5, contador de la division
152           ADD    R7, R0, #1      ; E2807001 ; Movemos un 1 al registro R7, acumulador para division
153           SUBS   R5, R5, R8      ; E0555008 ; Corregimos R5 segun la cantidad de luces que hayamos encendido y actualiza banderas
154           BEQ    salto          ; 0A000002 ; En caso de ser 0, debemos evitar el loop division
155
156 division:  SUBS   R5, R5, #1      ; E2555001 ; Restamos uno al contador y actualizamos banderas
157           ADD    R7, R7, R7      ; E0877007 ; Acumulamos R7
158           BNE    division       ; 1AFFFFFC ;
159
160 salto:     ADD    R8, R8, #1      ; E2888001 ; Sumamos 1 al registro R8, correspondiente a la luz que sacamos
161           SUB    R6, R6, R7      ; E0466007 ; Division por 2 de R6
162           LDR    R9, [R0, #0x4]   ; E5909004 ; Lectura del delay desde tabla, guardamos en el registro R9
163
164 delay      SUBS   R9, R9, #1      ; E2599001 ; Restamos uno al contador y actualizamos banderas
165           BNE    delay          ; 1AFFFFFD ;
166
167           SUBS   R4, R4, #1      ; E2544001 ; Restamos uno al contador del loop1
168           BNE    loop1          ; 1AFFFFF0 ;
169
170
171           ADD    R3, R0, #0x2     ; E2803002 ; Movemos un 2 al registro R2, contador para el bucle de lectura
172
173 lectura:   SUBS   R3, R3, #1      ; E2533001 ; Restamos uno al contador y actualizamos banderas
174           LDR    R1, [R0, #0x800] ; E5901800 ; Leemos el puerto, guardando en R1 los valores de los switches y del key_1
175           BNE    lectura        ; 1AFFFFFC ;
176
177           ANDS   R1, R1, #0x02    ; E2111002 ; Primer mascara, verificamos que el pin T8 se haya apagado
178           BEQ    inicio          ; 0AFFFF91 ;
179
180 loop2:     ADD    R6, R6, R6      ; E0866006 ; Multiplicamos R6 por 2
181           LDR    R9, [R0, #0x4]   ; E5909004 ; Lectura del delay desde tabla, guardamos en el registro R9
182
183 delay      SUBS   R9, R9, #1      ; E2599001 ; Restamos uno al contador y actualizamos banderas
184           BNE    delay          ; 1AFFFFFD ;
185
186           STR    R6, [R0, #0x800] ; E5806800 ; Sacamos la luz al puerto
187           SUBS   R4, R4, #1      ; E2544001 ; Restamos uno al contador del loop2
188           BNE    loop2          ; 1AFFFFF8 ;
189
190           ADD    R3, R0, #0x2     ; E2803002 ; Movemos un 2 al registro R2, contador para el bucle de lectura
191
192 lectura:   SUBS   R3, R3, #1      ; E2533001 ; Restamos uno al contador y actualizamos banderas
193           LDR    R1, [R0, #0x800] ; E5901800 ; Leemos el puerto, guardando en R1 los valores de los switches y del key_1
194           BNE    lectura        ; 1AFFFFFC ;
195
196           ANDS   R1, R1, #0x02    ; E2111002 ; Segunda mascara, verificamos que el pin T8 se haya apagado
197           BEQ    inicio          ; 0AFFFF84 ;
198
199           BNE    sec2           ; 1AFFFFD7 ; De no haberse apagado el pin T8, se reinicia la secuencia
200

```



```

202 /-----/
203 /----- Secuencia 3 -----/
204 /-----/
205
206 sec3:      ADD r4, r0, #0x1f      ; E280401F ; Contador de palabras de la tabla
207          ADD r5, r0, #44         ; E2805028 ; Guardamos el valor de la primer palabra de la secuencia
208
209 loop1:     LDR r2, [r5]           ; E5952000 ; Lee el dato de la tabla guardado en r2
210          STR r2, [r0, #0x800]    ; E5802800 ; Sacamos la luz al puerto del LED
211          LDR r9, [r0, #4]        ; E5909004 ; Cargamos en el registro el valor de tabla del delay
212
213 delay:     SUBS r9, r9, #1        ; E2599001 ; Resta 1 hasta llegar a cero
214          BNE delay              ; 1AFFFFF0 ;
215
216          ADD r5, r5, #4          ; E2855004 ; Aumentamos en 4 para pasar a la proxima palabra de tabla
217          SUBS r4, r4, #1         ; E2544001 ; Restamos 1 al contador de instrucciones
218          ADDEQ r4, r0, #0x1f     ; 0280401F ; Si llega a cero, suma la cantidad de líneas de la tabla
219          ADDEQ r5, r0, #44       ; 02805028 ; Y vuelve a la primer palabra de la secuencia
220          ADD R3, R0, #0x2        ; E2803002 ; Contador de doble lectura
221
222 lectura:  SUBS R3, R3, #1        ; E2533001 ; Lectura del puerto
223          LDR R1, [R0, #0x800]    ; E5901800 ;
224          BNE lectura             ; 1AFFFFF0 ;
225
226          ANDS R1, R1, #0x04      ; E2111004 ; Mascara: verifica que se haya apagado el pin_B9
227          BEQ inicio              ; 0AFFFFF2 ;
228          BNE loop1              ; 1AFFFFF0 ; Si no se apaga el PIN_B9 vuelve a la secuencia

```




```

230 /-----/
231 /----- Secuencia 4 -----/
232 /-----/
233
234 sec4:      ADD    R4, R0, #5      ; E2804005 ; Movemos un 5 al registro R5, contador del loop1
235           ADD    R8, R0, #0      ; E2808000 ; Movemos un 0 al registro R8, contador de luces encendidas
236           ADD    R6, R0, #0x80   ; E2806080 ; Movemos el valor 0x80 al registro R6, sera nuestro registro de luces
237
238 loop1:     STR    R6, [R0, #0x80] ; E5806800 ; Sacamos luz al puerto
239           ADD    R5, R0, #6      ; E2805006 ; Movemos un 6 al registro R5, contador de la division
240           ADD    R7, R0, #1      ; E2807001 ; Movemos un 1 al registro R7, acumulador de la division
241           SUB    R5, R5, R8      ; E0455008 ; Corregimos R5 segun la cantidad de luces que hemos encendido
242
243 division:  SUBS   R5, R5, #1      ; E2555001 ; Restamos uno al contador y actualizamos banderas
244           ADD    R7, R7, R7      ; E0877007 ; Acumulamos R7
245           BNE    division        ; 1AFFFFFC ;
246
247           ADD    R8, R8, #1      ; E2888001 ; Sumamos uno a R8, correspondiente a la luz encendida
248           SUB    R6, R6, R7      ; E0466007 ; Division de R6 por 2
249           LDR    R9, [R0, #0x4]  ; E5909004 ; Lectura del delay desde tabla, guardamos en el registro R9
250
251 delay      SUBS   R9, R9, #1      ; E2599001 ; Restamos uno al contador y actualizamos banderas
252           BNE    delay          ; 1AFFFFFD ;
253
254           SUBS   R4, R4, #1      ; E2544001 ; Restamos uno al contador del loop y actualizamos banderas
255           BNE    loop1          ; 1AFFFFF1 ;
256
257           ADD    R6, R0, #0x08   ; E2806008 ; Movemos a R6 el valor 0x08, correspondiente al cuarto led menos significativo
258           ADD    R4, R0, #3      ; E2804003 ; Movemos un 3 al registro R4, contador del loop2
259           ADD    R10, R6, R6     ; E086A006 ; Multiplicamos R6 por 2 y lo guardamos en R10
260           ADD    R8, R0, #0      ; E2808000 ; Volvemos R8 a 0, cumplira el mismo proposito
261           ADD    R3, R0, #0x2    ; E2803002 ; Movemos un 2 al registro R3, contador para el bucle de lectura
262
263 ~ lectura: SUBS   R3, R3, #1      ; E2533001 ; Restamos uno al contador y actualizamos banderas
264           LDR    R1, [R0, #0x800] ; E5901800 ; Leemos el puerto, guardando en R1 los valores de los switches y del key_1
265           BNE    lectura        ; 1AFFFFFC ;
266
267           ANDS   R1, R1, #0x08   ; E2111008 ; Primer mascara, verificamos que el pin_M15 se haya apagado
268           BEQ    inicio         ; 0AFFFF56 ;
269
270 ~ loop2:   ADD    R5, R0, #2      ; E2805002 ; Movemos un 2 al registro R5, contador de la division
271           ADD    R7, R0, #1      ; E2807001 ; Movemos un 1 al registro R7, acumulador de la division
272           SUBS   R5, R5, R8      ; E0555008 ; Corregimos R5 segun la cantidad de luces que hayamos encendido
273           BEQ    salto          ; 0A000002 ; En caso de ser 0, debemos evitar el loop division
274
275 ~ division: SUBS   R5, R5, #1      ; E2555001 ; Restamos uno al contador y actualizamos banderas
276           ADD    R7, R7, R7      ; E0877007 ; Acumulamos R7
277           BNE    division        ; 1AFFFFFC ;
278
279           ADD    R8, R8, #1      ; E2888001 ; Sumamos uno a R8, correspondiente a la luz encendida
280           SUB    R6, R6, R7      ; E0466007 ; Division de R6 por 2
281           ORR    R11, R10, R6    ; E18AB006 ; Mediante una OR logica, unimos las 2 señales para formar una sola
282           STR    R11, [R0, #0x800] ; E580B800 ; Sacamos luz al puerto
283           ADD    R10, R10, R10   ; E08AA00A ; Multiplicamos R10 por 2
284           LDR    R9, [R0, #0x8]  ; E5909004 ; Lectura del delay desde tabla, guardamos en el registro R9
285
286 delay      SUBS   R9, R9, #1      ; E2599001 ; Restamos uno al contador y actualizamos banderas
287           BNE    delay          ; 1AFFFFFD ;
288
289           SUBS   R4, R4, #1      ; E2544001 ; Restamos uno al contador del loop
290           BNE    loop2          ; 1AFFFFEE ;
291
292           ADD    R6, R0, #0x00   ; E2806000 ; Movemos un 0 al registro R6, correspondiente a la última luz de la secuencia
293           STR    R6, [R0, #0x800] ; E5806800 ; Sacamos luz al puerto
294
295           LDR    R9, [R0, #0x8]  ; E5909004 ; Lectura del delay desde tabla, guardamos en el registro R9
296
297 delay      SUBS   R9, R9, #1      ; E2599001 ; Restamos uno al contador y actualizamos banderas
298           BNE    delay          ; 1AFFFFFD ;
299
300           ADD    R3, R0, #0x2    ; E2803002 ; Movemos un 2 al registro R3, contador para el bucle de lectura
301
302 lectura:   SUBS   R3, R3, #1      ; E2533001 ; Restamos uno al contador y actualizamos banderas
303           LDR    R1, [R0, #0x800] ; E5901800 ; Leemos el puerto, guardando en R1 los valores de los switches y del key_1
304           BNE    lectura        ; 1AFFFFFC ;
305
306           ANDS   R1, R1, #0x08   ; E2111008 ; Segunda mascara, verificamos que el pin_M15 se haya apagado
307           BEQ    inicio         ; 0AFFFF3D ;
308           BNE    sec4           ; 1AFFFFCA ; De no haberse apagado, repetimos la secuencia

```




Conclusiones

El trabajar con un lenguaje de bajo nivel como *Assembly* nos ha permitido observar más de cerca cómo funcionan los microprocesadores y qué operaciones llevan a cabo. Al realizar las distintas operaciones que nos permitían introducir secuencias a la placa, pudimos familiarizarnos un poco más con dicha complejidad de un lenguaje de bajo nivel. A su vez, dicha complejidad se vió reforzada por el uso de una ARM con operaciones reducidas, pero que al final logramos entender y encontrar las soluciones para sus distintas limitaciones, lo que nos permitió cumplir, en su gran mayoría, con los objetivos planteados al comienzo del trabajo.



Bibliografía

- Harris, S. L., & Harris, D. (2015). *Digital design and computer architecture*. Morgan Kaufmann.
- Terasic Technologies (2003). *DE0-Nano User Manual*.