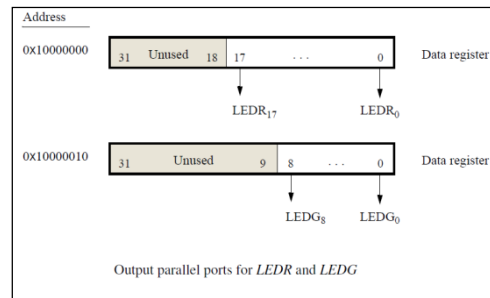


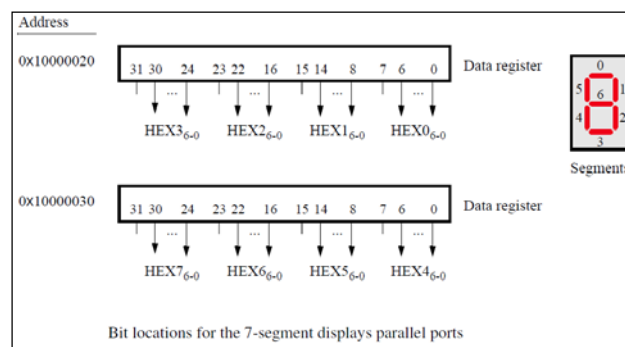
PRÁCTICA 2

EJERCICIOS DE PRÁCTICAS - ESTRUCTURA DE COMPUTADORES

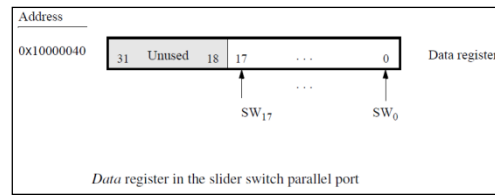
1. Diseñar un programa ensamblador que encienda alternativamente todos los leds rojos en posiciones pares y todos los leds rojos en posiciones impares de la placa DE2. Nota: observar la asignación de la dirección del espacio de direccionamiento del procesador NIOS II de la interfaz paralela de los leds rojos (LEDR) que se muestra a continuación.



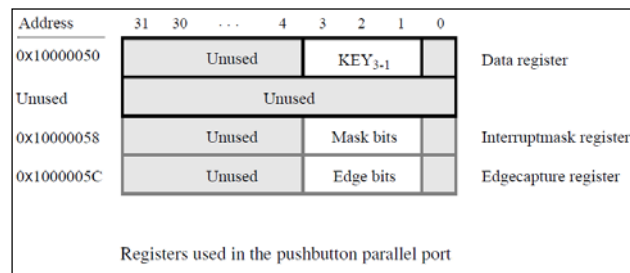
2. Diseñar un programa ensamblador que encienda alternativamente cada uno de los leds verdes de la placa DE2. Nota: observar la asignación de la dirección del espacio de direccionamiento del procesador NIOS II de la interfaz paralela de los leds verdes (LEDG) que se encuentra en el ejercicio anterior.
3. Diseñar un programa ensamblador que muestre en la placa DE2 un cero en el primer display "siete-segmentos" y lo rote por los cuatro primeros displays "siete-segmentos". Nota: observar las asignaciones de direcciones de las interfaces paralelas de los displays siete-segmentos que se muestran a continuación.



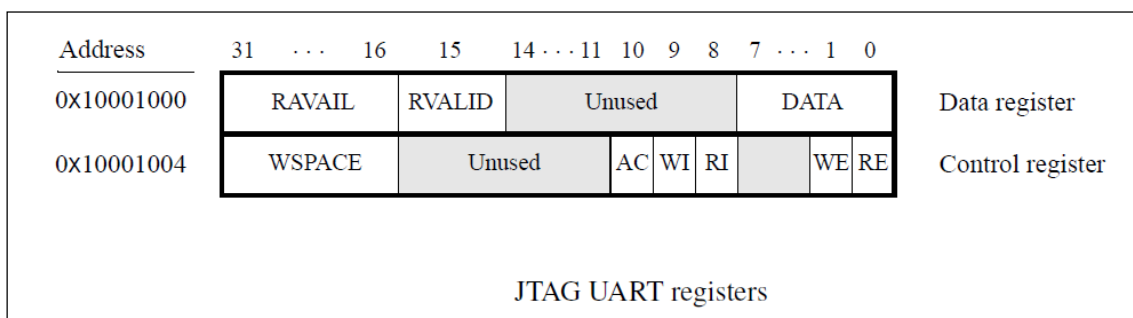
4. Diseñar un programa ensamblador que encienda en la placa DE2 los leds rojos correspondientes a los interruptores que no estén activados. Nota: observar la asignación de direcciones de la interfaz paralela de los interruptores que se muestran a continuación, así como la de los leds rojos que se muestra en los ejercicios 1 y 2.



5. Diseñar un programa ensamblador del procesador NIOS II que muestre en la placa DE2 un 0 en el primer display 7-segmentos y luego lo sustituya por 1, 2 ó 3 en función de la pulsación de los botones correspondientes en la placa DE2. Nota: observar la asignación de direcciones de la interfaz paralela de los botones de DE2 que se muestran a continuación, así como la de los displays 7-segmentos que se muestra en el ejercicio 3.



6. Diseñar un programa ensamblador que, utilizando la interfaz serie JTAG-UART de la placa DE2, haga aparecer en la terminal del Altera Monitor cualquier carácter que sea tecleado en la misma. Este programa debe modificar las letras mayúsculas para que aparezcan en la terminal como minúsculas y viceversa. El resto de caracteres debe aparecer sin modificar. Nota-1: observar la asignación de la dirección del registro de datos de la interfaz JTAG-UART y los campos en los que se divide este registro. Su bit 15 (RVALID) indica si la interfaz dispone en el campo DATA de un nuevo byte que no se ha leído previamente (RVALID=1). Nota-2: la cantidad de espacio disponible en la FIFO de la interfaz JTAG-UART se refleja en los bits 16 a 31 del registro de control (WSPACE). Si WSPACE=0, significa que no queda espacio en la FIFO para guardar más caracteres. Nota-3: en ASCII, las mayúsculas se encuentran en el rango de caracteres 65..90, y las minúsculas se encuentran en el rango 97..122.



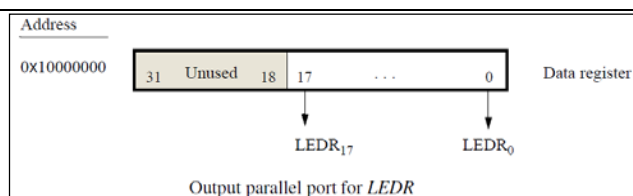
0	<NUL>	32	<SPC>	64	@	96	`	128	À	160	†	192	¿	224	‡
1	<SOH>	33	!	65	A	97	a	129	Á	161	°	193	¡	225	·
2	<STX>	34	"	66	B	98	b	130	Â	162	¢	194	ª	226	,
3	<ETX>	35	#	67	C	99	c	131	Ã	163	£	195	»	227	"
4	<EOT>	36	\$	68	D	100	d	132	Ä	164	§	196	ƒ	228	‰
5	<ENQ>	37	%	69	E	101	e	133	Å	165	•	197	≈	229	Å
6	<ACK>	38	&	70	F	102	f	134	Ü	166	¶	198	Δ	230	Ê
7	<BEL>	39	'	71	G	103	g	135	á	167	ß	199	«	231	Á
8	<BS>	40	(72	H	104	h	136	à	168	®	200	»	232	È
9	<TAB>	41)	73	I	105	i	137	â	169	©	201	...	233	É
10	<LF>	42	*	74	J	106	j	138	ä	170	™	202		234	Í
11	<VT>	43	+	75	K	107	k	139	å	171	‘	203	Ä	235	Î
12	<FF>	44	,	76	L	108	l	140	ä	172	”	204	Å	236	Ï
13	<CR>	45	-	77	M	109	m	141	ç	173	≠	205	Ö	237	İ
14	<SO>	46	.	78	N	110	n	142	é	174	/E	206	Æ	238	Ó
15	<SI>	47	/	79	O	111	o	143	è	175	Ø	207	œ	239	Ô
16	<DLE>	48	0	80	P	112	p	144	ê	176	∞	208	-	240	•
17	<DC1>	49	1	81	Q	113	q	145	ë	177	±	209	—	241	Ö
18	<DC2>	50	2	82	R	114	r	146	í	178	≤	210	"	242	Ú
19	<DC3>	51	3	83	S	115	s	147	ì	179	≥	211	"	243	Û
20	<DC4>	52	4	84	T	116	t	148	î	180	¥	212	`	244	Ü
21	<NAK>	53	5	85	U	117	u	149	ï	181	µ	213	'	245	ı
22	<SYN>	54	6	86	V	118	v	150	ñ	182	ð	214	÷	246	ˆ
23	<ETB>	55	7	87	W	119	w	151	ó	183	Σ	215	◊	247	˜
24	<CAN>	56	8	88	X	120	x	152	ò	184	Π	216	ŷ	248	—
25		57	9	89	Y	121	y	153	ô	185	π	217	Ÿ	249	˘
26	<SUB>	58	:	90	Z	122	z	154	ö	186	ƒ	218	/	250	˙
27	<ESC>	59	;	91	[123	{	155	õ	187	ª	219	€	251	º
28	<FS>	60	<	92	\	124		156	ú	188	º	220	<	252	˚
29	<GS>	61	=	93]	125	}	157	ù	189	Ω	221	>	253	˛
30	<RS>	62	>	94	^	126	~	158	û	190	æ	222	fi	254	˜
31	<US>	63	?	95	_	127		159	ü	191	ø	223	fl	255	˘

7. Utilizando interrupciones, diseñar un programa ensamblador del NIOS II que encienda alternativamente, de uno en uno, todos los leds rojos de la placa DE2. Inicialmente, el movimiento de encendido será hacia la izquierda comenzando por el led rojo situado en la posición menos significativa. Cuando se pulse el botón KEY1 la dirección del movimiento cambiará hacia la derecha. Cuando se pulse el botón KEY2 la dirección de movimiento cambiará de nuevo hacia la izquierda.

En el diseño del programa deben utilizarse dos tipos de interrupciones: las del temporizador y las del panel de botones. Por un lado, el temporizador será configurado para lanzar una interrupción periódicamente que indicará que debe realizarse un desplazamiento en los leds. Por otro lado, el panel de botones será configurado para lanzar una interrupción cuando se pulse un botón.

I/O Peripheral	IRQ #
Interval timer	0
Pushbutton switch parallel port	1
JTAG port	8
Serial port	10
JP1 Expansion parallel port	11
JP2 Expansion parallel port	12

Hardware IRQ interrupt assignment for the DE2 Basic Computer



Register	Name	Function
r0	zero	0x00000000
r1	at	Assembler Temporary
r2		
r3		
.	.	.
.	.	.
.	.	.
r23		
r24	et	Exception Temporary (1)
r25	bt	Breakpoint Temporary (2)
r26	gp	Global Pointer
r27	sp	Stack Pointer
r28	fp	Frame Pointer
r29	ea	Exception Return Address (1)
r30	ba	Breakpoint Return Address (2)
r31	ra	Return Address

General Purpose registers

Address	31	...	17	16	15	...	3	2	1	0						
0x10002000	Not present (interval timer has 16-bit registers)								Unused		RUN	TO	Status register			
0x10002004									Unused		STOP	START	CONT	ITO	Control register	
0x10002008									Counter start value (low)							
0x1000200C									Counter start value (high)							
0x10002010									Counter snapshot (low)							
0x10002014									Counter snapshot (high)							

Interval timer registers

Register	Name	b ₃₁	...	b ₂	b ₁	b ₀
ctl0	status	Reserved			U	PIE
ctl1	estatus	Reserved			EU	EPIE
ctl2	bstatus	Reserved			BU	BPIE
ctl3	ienable	Interrupt-enable bits				
ctl4	ipending	Pending-interrupt bits				
ctl5	cpuid	Unique processor identifier				

Control registers

Address	31	30	...	4	3	2	1	0	
0x10000050	Unused				KEY ₃₋₁				Data register
Unused	Unused								
0x10000058	Unused				Mask bits				Interruptmask register
0x1000005C	Unused				Edge bits				Edgecapture register

Registers used in the pushbutton parallel port

8. Selecciona de la lista adjunta las instrucciones que se deben incluir en las posiciones 4, 6, 10 y 21 del programa en ensamblador para que se pueda ejecutar correctamente en DE2. El programa realiza la siguiente funcionalidad: *enciende alternativamente todos los leds rojos en posiciones pares y todos los leds rojos en posiciones impares de la placa DE2.*

Lista de instrucciones:

- a) movia r4, 500000
- b) INI: ldw r3, ROJOPAR(r0)
- c) bne r2,r0,LOOP
- d) subi r4,r4,1
- e) .word 0x2AAAA
- f) ldw r3, ROJOIMPAR(r0)

Programa:

```

1. .global _start
2. _start:
3. movia r2, 0x10000000
4.
5. stwio r3, 0(r2)
6.
7. DELAY1:
8. subi r4, r4, 1
9. bne r4, r0, DELAY1
10.
11. stwio r3, 0(r2)
12. movia r4, 500000

```

```

13. DELAY2:
14. subi r4, r4, 1
15. bne r4, r0, DELAY2
16. br INI
17. .org 0x200
18. ROJOPAR:
19. .word 0x15555
20. ROJOIMPAR:
21.
22. .end

```

9. Reordena la lista adjunta las instrucciones y directivas de ensamblador para que el programa resultado se pueda ejecutar correctamente en DE2 y realice la siguiente funcionalidad: *enciende alternativamente todos los leds rojos en posiciones pares y todos los leds rojos en posiciones impares de la placa DE2.*

Lista de instrucciones y directivas de ensamblador:

- a) .end
- b) .global _start
- c) .org 0x200
- d) .word 0x15555
- e) .word 0x2AAAA
- f) _start:
- g) bne r4, r0, DELAY1
- h) bne r4, r0, DELAY2
- i) br INI
- j) DELAY1:
- k) DELAY2:
- l) INI: ldw r3, ROJOPAR(r0)
- m) ldw r3, ROJOIMPAR(r0)
- n) movia r2, 0x10000000
- o) movia r4, 500000
- p) movia r4, 500000
- q) ROJOIMPAR:
- r) ROJOPAR:
- s) stwio r3, 0(r2)
- t) stwio r3, 0(r2)
- u) subi r4, r4, 1
- v) subi r4, r4, 1

Programa:

- 1. _____
- 2. _____
- 3. _____

4. _____
5. _____
6. _____
7. _____
8. _____
9. _____
10. _____
11. _____
12. _____
13. _____
14. _____
15. _____
16. _____
17. _____
18. _____
19. _____
20. _____
21. _____
22. _____