

Código de test de las IRQ en el P1

formato aritméticas:

ADD RD, RS, RT

formato lw, sw, beq:

LW RT, INM(RS)

SW RT, INM(RS)

BEQ RS, RT

Contenido Memoria Datos: [256, 1, 8, Nint (0 al empezar), 0, 0, 0, 0, 0...]

Pseudo-código

```
const int sp_init = 256;
const int one = 1;
const int eight = 8;
volatile static int Nint = 0;

void main(){
    register int R1;
    Print(sp_init); // Escribe un entero en el registro de salida
    while(1) {R1=2*R1;}}

void RTI (void) __irq { //__irq indica que es una excepción y que hay que
    Nint++;              // que generar el prólogo y epílogo adecuado
    Print(Nint);} // Escribe un entero en el registro de salida

void Abort (void) __abort{//__abort indica que es una excepción y que hay
    while(1){};} // que generar el prólogo y epílogo adecuado

void UNDEF (void) __undef {//__undef indica que es una excepción y que hay
    while(1){};} // que generar el prólogo y epílogo adecuado
```

Valores finales

r1=1,2,4,8,16....

Mem(C)= 0,1,2,3,4.....

Reset	@0x0	10210003	beq R1, R1, INI;	Se salta siempre a la @16 donde empieza el programa (@0x8)
IRQ	@0x4	1021003E	beq R1, R1, RTI;	Se salta siempre a la @64*4
DAbort	@0x8	1021005D	beq R1, R1, RT_Abort;	Se salta siempre a la @96*4
UNDEF	@0xC	1021006C	beq R1, R1, RT_UNDEF;	Se salta siempre a la @112*4
INI:	@0x10	081F0000	LW R31, 0(R0)	R31=Mem(0) = 256; R31 es el puntero de pila (SP)
		08010004	LW R1, 4(R0)	R1=Mem(4) = 1;
		83E00000	WRO R31	IO_output <= R1
Main		04210800	ADD R1, R1, R1	R1=2*R1
		1021FFFE	beq R1, R1, main	Bucle infinito. Solo se sale si hay una IRQ
RTI:	@0x100	0FE10000	SW R1, 0(R31)	Guardamos el contenido de R1 en pila
		0FE20004	SW R2, 4(R31)	Guardamos el contenido de R2 en pila
		08010008	LW R1, 8(R0)	R1=Mem(8) = 8;
		07E1F800	ADD r31, R1, R31	R31=R31 +8; Incrementamos el SP
		0802000C	LW R2, C(R0)	R2=Mem(C) En esta posición de memoria contabilizamos el número de int (Nint)

		08010004	LW R1, 4(R0)	$R1 = Mem(4) = 1;$
		04221000	ADD R2, R1, R2	$R2 = Nint++$
		80400000	WRO R2	$IO_output \leq R2$
		0C02000C	SW R2, C(R0)	$Mem(C) = Nint++$
		08010008	LW R1, 8(R0)	$R1 = Mem(8) = 8;$
		07E1F801	SUB r31, R31, R1	$SP = SP - 8$
		0BE10000	LW R1, 0(R31)	<i>Restauramos el contenido de R1 de pila</i>
		0BE20004	LW R2, 4(R31)	<i>Restauramos el contenido de R2 de pila</i>
		20000000	rte	<i>Se vuelve a la instrucción que se interrumpió</i>
RT_Abort	@0x180	1000FFFF	beq R0, R0, RT_Abort	
RT_UNDEF	bucleU	1000FFFF	beq R0, R0, RT_UNDEF	

