Código retardado para funcionar en la versión incompleta

El código está pensado para funcionar en un procesador con saltos 1-retardados que no detecta los riesgos de datos

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, 0, 0, 0, 0, 0…]

Pseudo-código

```
const int dato1 = 256;
const int dato2 = 1;
int resultado = 8;

void main(){
      resultado = dato1+dato2;
      while(1){}}
```

Valores finales

r1=256; r2= 1; r3=257; r31=256;
Mem(8)= 257

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)*
Main:	@0x14	08010000	LW R1, 0(R0)	*R1=M(0)=256*
	@0x18	08020004	LW R2, 4(R0)	*R2=M(4)=1*
	@0x1C	00000000	nop	*Para evitar riesgos de datos*
	@0x20	00000000	nop	
	@0x24	04221800	ADD R3, R1, R2	*R3=R1+R2=257*
	@0x28	00000000	nop	*Para evitar riesgos de datos*
	@0x2C	00000000	nop	
	@0x30	0C030008	SW R3, 8(R0)	*M(8)=257*
End:	@0x34	1000FFFa	beq r0, r0, end	*Bucle infinito.*
	@0x38	00000000	nop	*Para evitar riesgos de control*