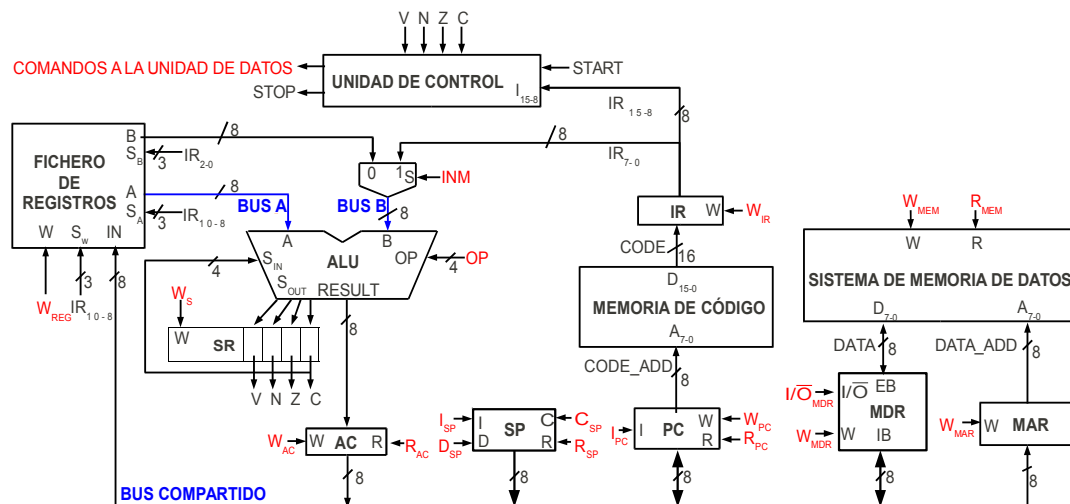


Nombre (en mayúsculas):

Apellidos (en mayúsculas):

## Problema 1 (5 puntos)

Para implementar el CS2010 se propone la unidad de datos vista en clase:



OP <sub>3</sub>	OP <sub>2</sub>	OP <sub>1</sub>	OP <sub>0</sub>	RESULT=	V <sub>OUT</sub> =	N <sub>OUT</sub> =	Z <sub>OUT</sub> =	C <sub>OUT</sub> =
0	0	-	0	-	V <sub>IN</sub>	N <sub>IN</sub>	Z <sub>IN</sub>	0
0	0	0	1	-	-	-	-	-
0	0	1	1	-	V <sub>IN</sub>	N <sub>IN</sub>	Z <sub>IN</sub>	1
0	1	0	0	SHR(A, C <sub>IN</sub> )	C <sub>IN</sub> EXOR A <sub>0</sub>	RESULT <sub>7</sub>	NOT OR <sub>i=0</sub> <sup>7</sup> (RESULT <sub>i</sub> )	A <sub>0</sub>
0	1	0	1	SHL(A, C <sub>IN</sub> )	A <sub>7</sub> EXOR A <sub>6</sub>	RESULT <sub>7</sub>	NOT OR <sub>i=0</sub> <sup>7</sup> (RESULT <sub>i</sub> )	A <sub>7</sub>
0	1	1	-	A	-	-	-	-
1	0	0	-	(A + B) mod 2 <sup>8</sup>	overflow(A+B)	RESULT <sub>7</sub>	NOT OR <sub>i=0</sub> <sup>7</sup> (RESULT <sub>i</sub> )	carry(A+B)
1	0	1	-	(A - B) mod 2 <sup>8</sup>	underflow(A-B)	RESULT <sub>7</sub>	NOT OR <sub>i=0</sub> <sup>7</sup> (RESULT <sub>i</sub> )	borrow(A-B)
1	1	-	-	B	-	-	-	-

descripción de la ALU:

Se desea añadirle las instrucciones descritas en la tabla inferior. Su formato de instrucción no tiene porque coincidir con los propuestos en clase. Para cada una de ellas se pide lo siguiente:

- Indique si la instrucción puede ser llevada a cabo con la unidad de datos propuesta.
- Si la respuesta al apartado anterior es afirmativa, indique la secuencia de microoperaciones que requiere la fase de ejecución de la instrucción y los bits del código de instrucción que codificarán los números de los registros empleados como operandos (i, k).

instrucción	POP Ri	STDNEXT Ri,(Rk)
semántica	$R_i \leftarrow \text{data\_mem}[SP+1]; SP \leftarrow SP+1$	$\text{data\_mem}[R_k] \leftarrow R_i; R_k \leftarrow R_k+1$
¿puede realizarse?		
codificación	<u>I<sub>15</sub></u> <u>I<sub>14</sub></u> <u>I<sub>13</sub></u> <u>I<sub>12</sub></u> <u>I<sub>11</sub></u> <u>I<sub>10</sub></u> <u>I<sub>9</sub></u> <u>I<sub>8</sub></u> <u>I<sub>7</sub></u> <u>I<sub>6</sub></u> <u>I<sub>5</sub></u> <u>I<sub>4</sub></u> <u>I<sub>3</sub></u> <u>I<sub>2</sub></u> <u>I<sub>1</sub></u> <u>I<sub>0</sub></u>	<u>I<sub>15</sub></u> <u>I<sub>14</sub></u> <u>I<sub>13</sub></u> <u>I<sub>12</sub></u> <u>I<sub>11</sub></u> <u>I<sub>10</sub></u> <u>I<sub>9</sub></u> <u>I<sub>8</sub></u> <u>I<sub>7</sub></u> <u>I<sub>6</sub></u> <u>I<sub>5</sub></u> <u>I<sub>4</sub></u> <u>I<sub>3</sub></u> <u>I<sub>2</sub></u> <u>I<sub>1</sub></u> <u>I<sub>0</sub></u>
microoperación 1		
microoperación 2		
:		

## Problema 2 (5 puntos)

Definimos la función natural de variable natural  $f$  de la forma siguiente:

$$f(n) = \sum_{i=1}^n i$$

Una aplicación determinada escrita en ensamblador de AVR requiere usar una subrutina que compute  $f$ . Dicha subrutina debe recibir en R2 como entrada un número natural  $n$  y debe devolver codificado  $f(n)$  en la concatenación R1 y R0 (en R1 la parte más significativa y en R0 la menos significativa). Se pide lo siguiente:

- Escriba una subrutina que haga el cómputo de la función realizando las  $n-1$  sumas sucesivas  $1+2+\dots+n$ .
- Escriba otra versión de la subrutina que relice el cómputo más rápido aplicando la siguiente igualdad:

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}$$

Las subrutinas deben dejar los registros de propósito general tal y como estaban cuando se las llamó, salvo los mencionados R0, R1 y R2.