

**Apellido y Nombre**

email:

Fecha: 18-12-2019

Padrón:

### Cuatr. de cursada:

### Turno de TP:

### Turno de TP:

- 1) (a) Explique cuáles son los componentes de la microarquitectura ARC que hacen que la ALU 'vea' en cada entrada sólo uno de todos los registros del procesador (este es el que fue elegido por el programador).  
 (b) Proponga un microcódigo para la instrucción *jmpl*, asignándole ubicación en la memoria de control. Debería estar repetido en varias localizaciones?  
 (c) Explique de qué forma los flags de la ALU controlan el flujo del microprograma detallando los componentes circuitales que intervienen.
  - 2) Un programa recibe por stack la dirección y largo de un arreglo conformado por números enteros en complemento a 2 y se encarga de devolver por esa pila la suma de todos sus elementos positivos de dicho arreglo. Antes de terminar invoca una subrutina que escribe un 0 en la dirección A2010B10h en caso de que la suma anterior no sea representable en 32 bits o escribe un 1 en caso contrario. (a) subrutina y main forman parte del mismo módulo (b) indicar los cambios necesarios para que la rutina esté declarada en un módulo aparte
  - 3) Presente la tabla de símbolos del programa anterior (punto (a)) y explique detalladamente por qué esta información resulta fundamental para el correcta operación de la aplicación.
  - 4) (a) ¿Qué entiende por mapa de memoria? (b) Queda este definido al elegir un determinado microporcesador? (c) Explique eventuales diferencias entre los mapas de memoria de sistemas con y sin memoria cache

	Op			rd																op2																disp22																disp30															
SETHI Format	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Branch Format	0	0	0	cond				op2																				disp22																																							
CML format	0	1																disp30																																																	

Arithmetic Formats	1 0	rd	op3	rsl	0 0 0 0 0 0 0 0	rs2
	1 0	rd	op3	rsl	1	simm13
Memory Formats	1 1	rd	op3	rsl	0 0 0 0 0 0 0 0	rs2
	1 1	rd	op3	rsl	1	simm13

op	Format
00	SETHI/Branch
01	CALL
10	Arithmetic
11	Memory

op2	Inst.
010	branch
100	sethi

op3 (op=10)	
010000	addcc
010001	andcc
010010	orcc
010110	orncc
100110	srl
111000	jmpl

op3 (op=11)
000000 ld
000100 st

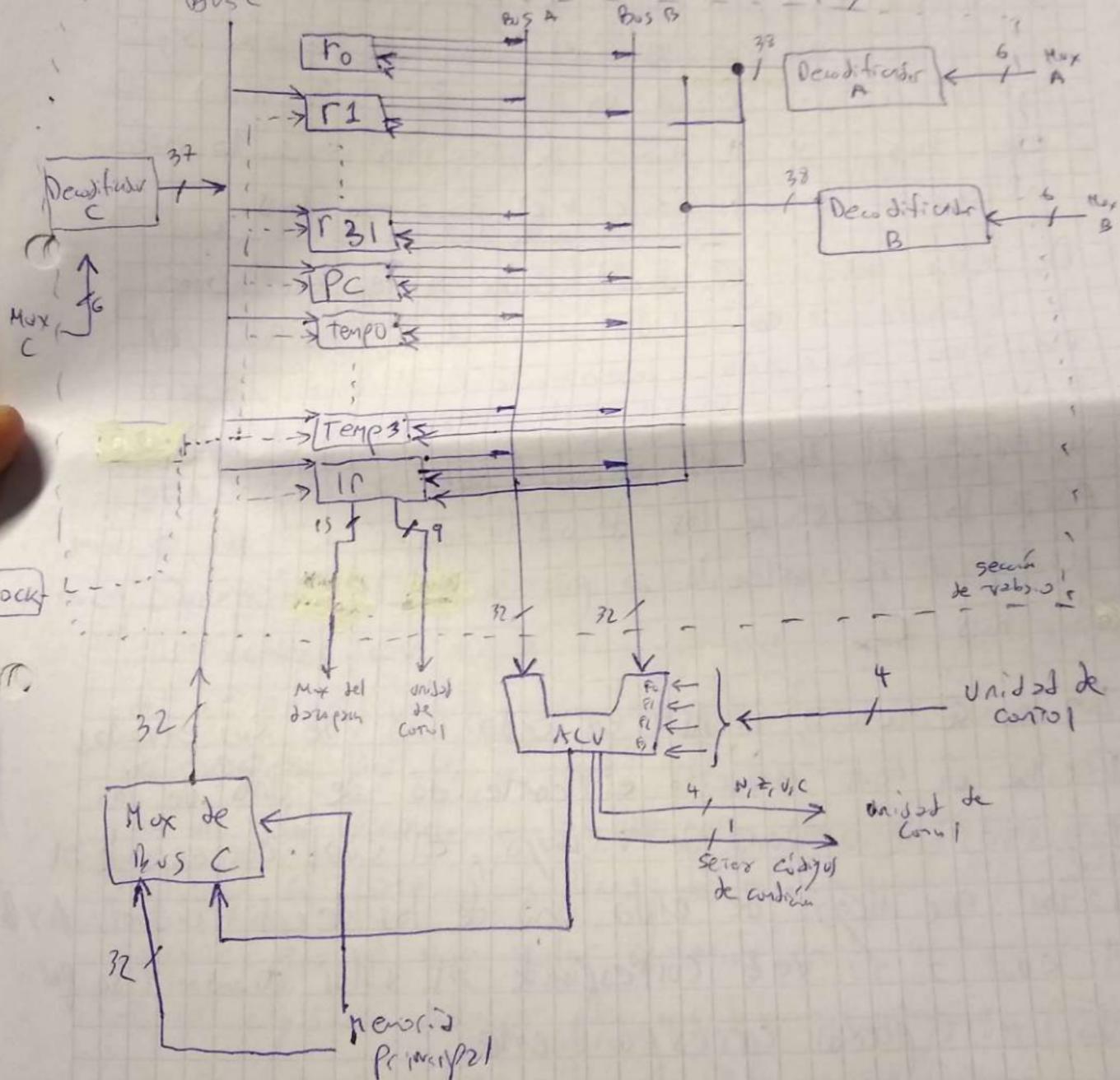
cond	branch
0001	be
0101	bcs
0110	bneq
0111	bvs
1000	ba

$F_3$	$F_2$	$F_1$	$F_0$	Operation
0	0	0	0	ANDC (A, B)
0	0	0	1	ORCC (A, B)
0	0	1	0	NORCC (A, B)
0	0	1	1	ADDC (A, B)
0	1	0	0	SRL (A, B)
0	1	0	1	AND (A, B)
0	1	1	0	OR (A, B)
0	1	1	1	NOR (A, B)
1	0	0	0	ADD (A, B)
1	0	0	1	LSHIFT2 (A)
1	0	1	0	LSHIFT10 (A)
1	0	1	1	SIMM13 (A)
1	1	0	0	SEXT13 (A)
1	1	0	1	INC (A)

cond	Operation
0 0 0	Use NEXT ADDR.
0 0 1	Use JUMP ADDR if $n = 1$
0 1 0	Use JUMP ADDR if $s = 1$
0 1 1	Use JUMP ADDR if $v = 1$
1 0 0	Use JUMP ADDR if $c = 1$
1 0 1	Use JUMP ADDR if $18[1..3] = 1$
1 1 0	Use JUMP ADDR
1 1 1	DECODE

[1]

a) Se detalla a continuación un gráfico que ~~expresa~~  
brinda una vista preliminar sobre lo preguntado.



Se explican los componentes del diagrama a continuación.  
Los Multiplexores para cada uno de los buses se encargan

