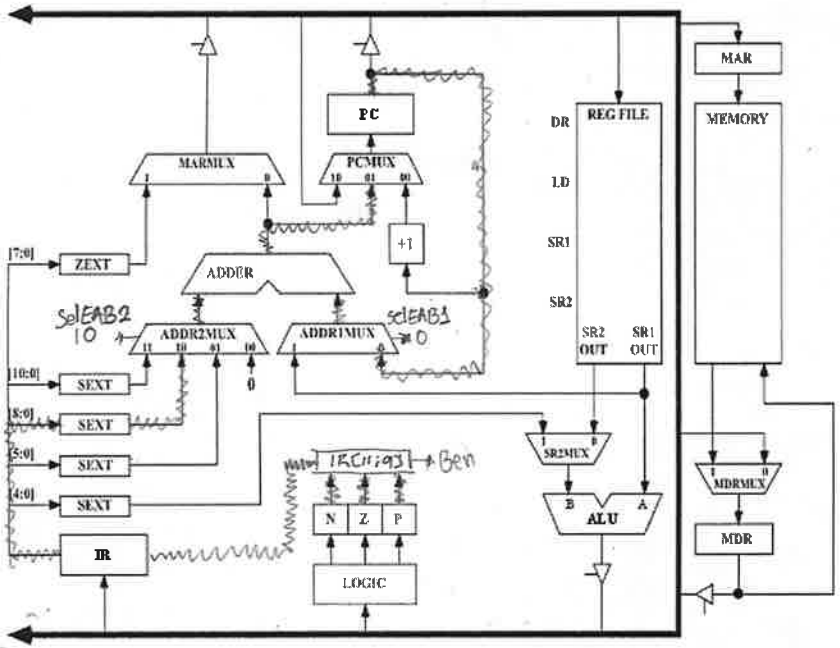
?

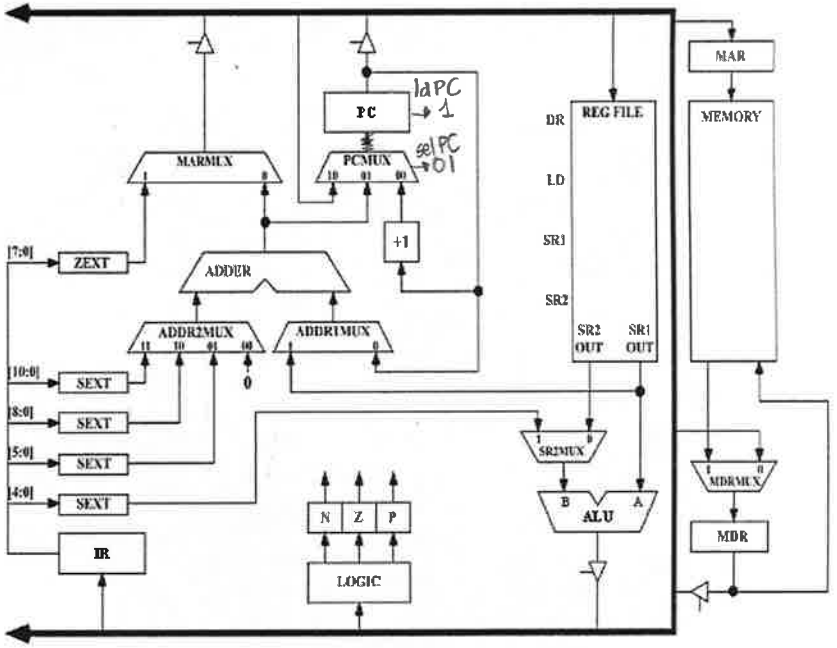
curState	IR[15:12]	IR[11]	nextState	aluControl	SR1	SR2	DR	selPC	selEAB1	selEAB2	enaALU	regWE	flagWE	enaMARM	selMAR	enaPC	ldPC	ldIR	ldMAR	ldMDR	selMDR	memWE	enaMDR	flagWE
fetch 0	xxxx	x	fetch 1	x	x xx	xxx	xxx									1			1					
fetch 1	xxxx	x	fetch 2					00									1		1		1			
fetch 2	xxxx	x	decode															1					1	
BR - decode	0000		branch																					
ADD - decode	0001		add																					
LD - decode	0010		load																					
ST - decode	0011		store																					
JSR,JSRR - decode	0100		jump to subroutine																					
AND - decode	0101		and																					
LDR - decode	0110		load relative																					
STR - decode	0111		store relative																					
RTI X decode	1000		return from interrupt																					
NOT - decode	1001		not																					
LDI - decode	1010		load indirect																					
STI - decode	1011		store indirect																					
JMP,RET - decode	1100		jump																					
X decode	1101		reserved																					
LEA decode	1110		load effective address																					
TRAP X decode	1111		system call TRAP																					
branch	xxxx	Ben 0	fetch 0						0	10														
branch	xxxx	Ben 1	branch 1						0	10														
branch 1	xxxx		fetch 0					01									1							
add	xxxx		fetch 0	01	IR[8:6]	IR[2:0]	IR[11:9]				1	1												1
load	xxxx		load 1						0	10				1	0				1					
load 1	xxxx		load 2																	1	1			
load 2	xxxx		fetch 0				IR[11:9]					1											1	1
store	xxxx		store 1						0	10				1	0				1					
store 1	xxxx		store 2	00	IR[11:9]						1									1	0			
store 2	xxxx		fetch 0																			1		
	xxxx																							

curState	IR[15:12]	IR[11]	nextState	aluControl	SR1	SR2	DR	selPC	selEAB1	selEAB2	enaALU	regWE	[illegible]	enaMARM	selMAR	enaPC	ldPC	ldIR	ldMAR	ldMDR	selMDR	memWE	enaMDR	flagWE
jump to Subroutine	x x x x		offset or baseR				7					1				1								
jump to Subroutine offset	x x x x	1	fetch 0					01	0	11							1							
jump to subroutine baseR	x x x x	0	fetch 0	00	IR[8:6]			10			1						1							
and	x x x x		fetch 0	10	IR[8:6]	IR[2:0]	IR[11:9]				1	1												1
load Relative	x x x x		load Relative 1		IR[8:6]				1	01				1	0				1					
load Relative 1	x x x x		load Relative 2																	1	1			
load Relative 2	x x x x		fetch 0				IR[11:9]					1											1	
store Relative	x x x x		store Relative 1		IR[8:6]				1	01				1	0				1					
store Relative 1	x x x x		store Relative 2	00	IR[11:9]						1									1	0			
store Relative 2	x x x x		fetch 0																			1		
not	x x x x		fetch 0	11	IR[8:6]		IR[11:9]				1	1												1
load Indirect	x x x x		load Indirect 1						0	10				1	0				1					
load Indirect 1	x x x x		load Indirect 2																	1	1			
load Indirect 2	x x x x		load Indirect 3																1				1	
load Indirect 3	x x x x		load Indirect 4																	1	1			
load Indirect 4	x x x x		fetch 0				IR[11:9]					1											1	
store Indirect	x x x x		store Indirect 1						0	10				1	0				1					
store Indirect 1	x x x x		store Indirect 2																	1	1			
store Indirect 2	x x x x		store Indirect 3																1				1	
store Indirect 3	x x x x		store Indirect 4	00	IR[11:9]						1									1	0			
store Indirect 4	x x x x		fetch 0																			1		
jump	x x x x		fetch 0	00	IR[8:6]			10			1						1							
load Effective Address	x x x x		fetch 0				IR[11:9]		0	10		1		1	0									

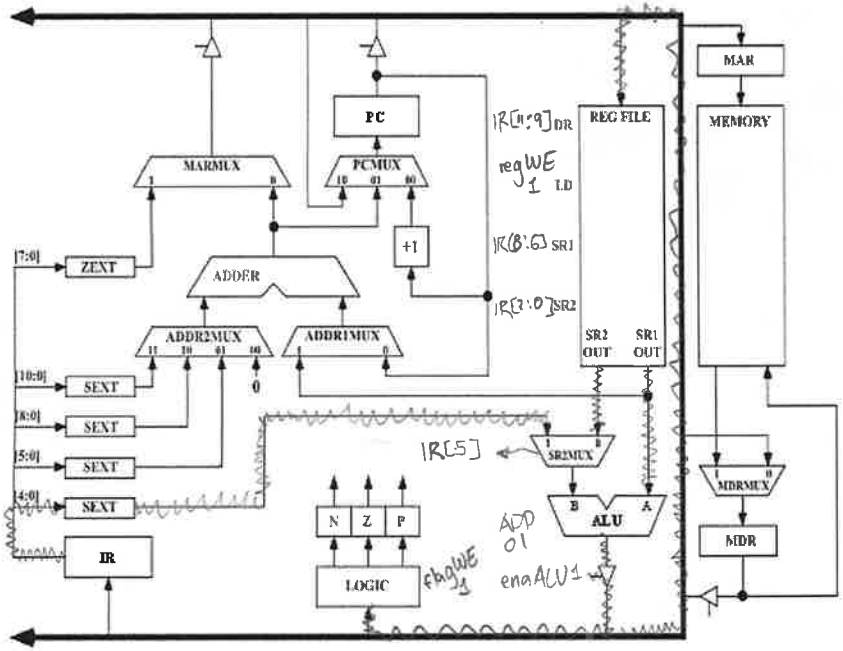
Branch



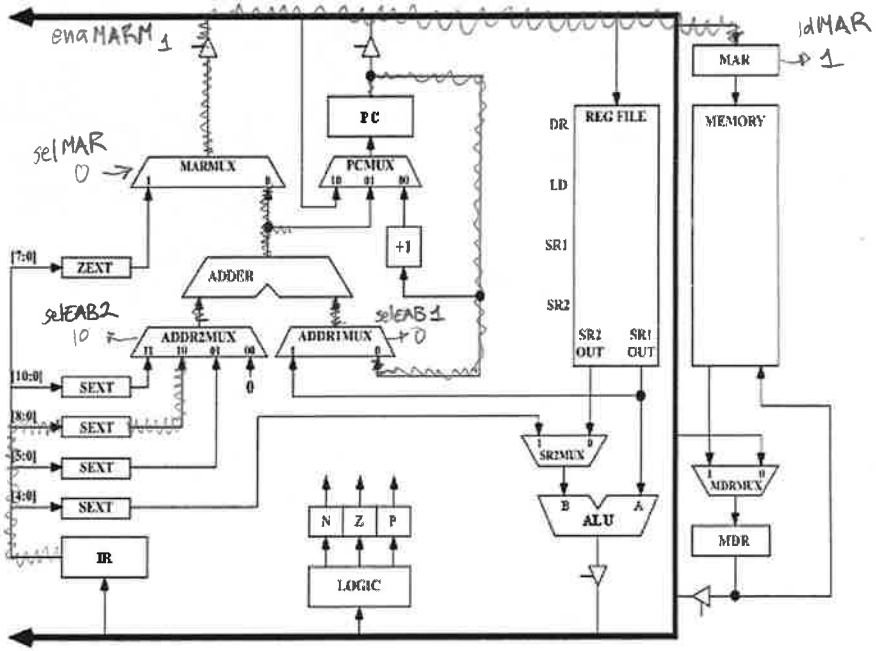
Branch 1



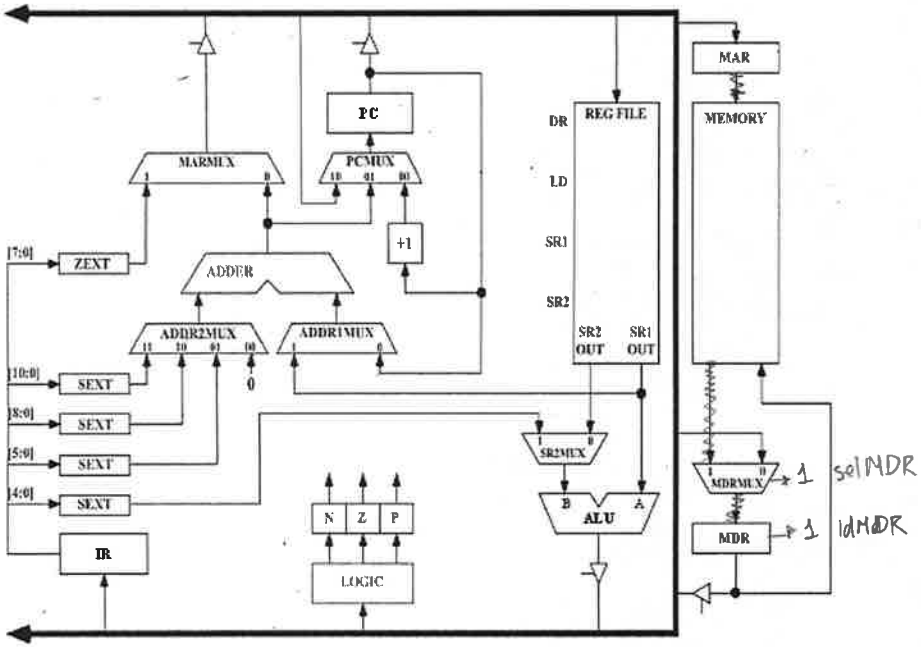
add



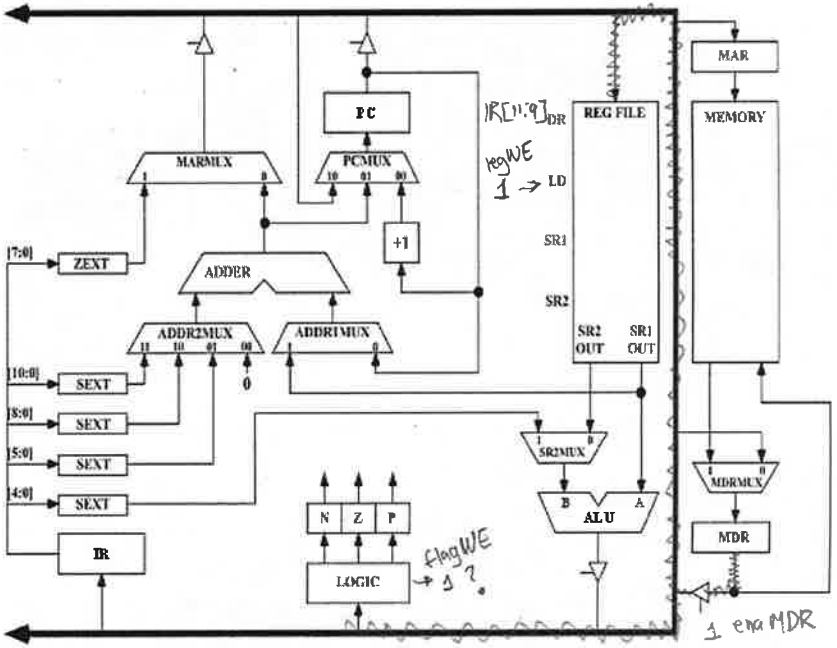
load



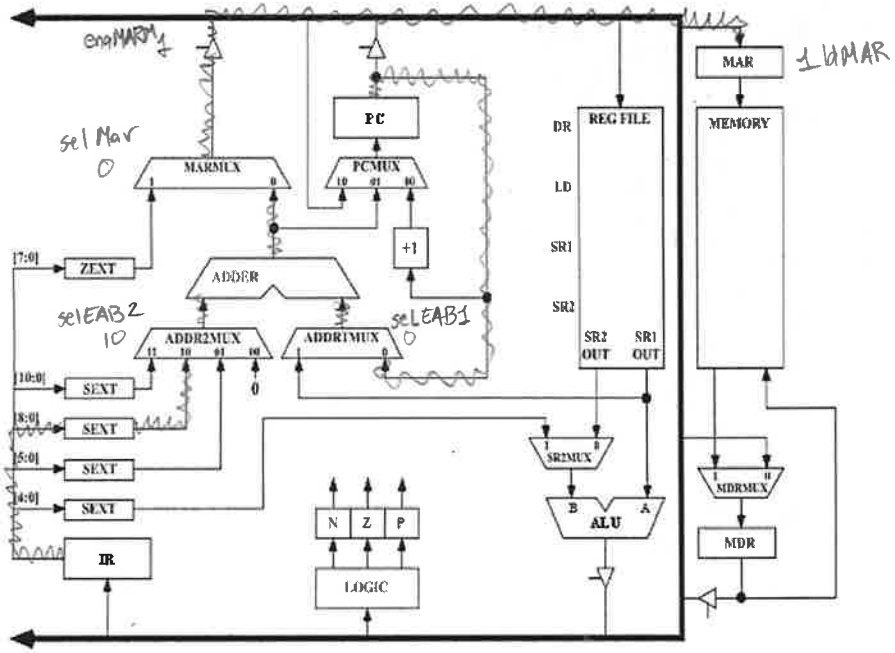
load 1



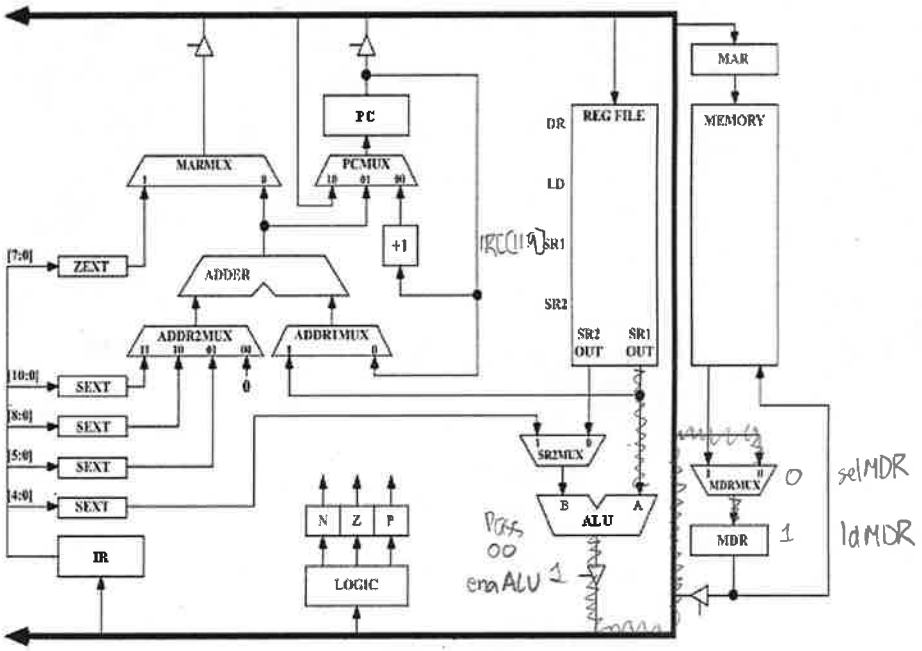
load 2



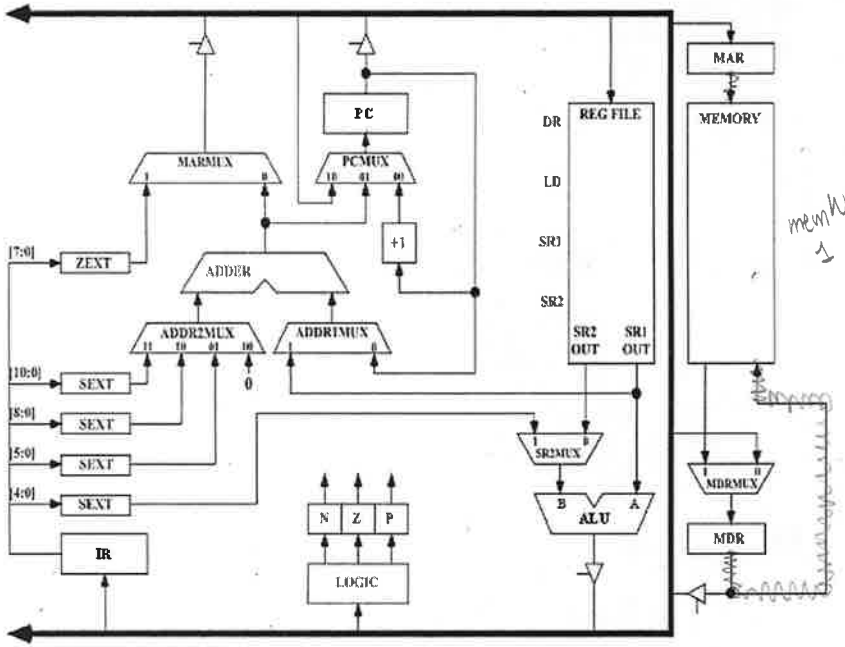
Store



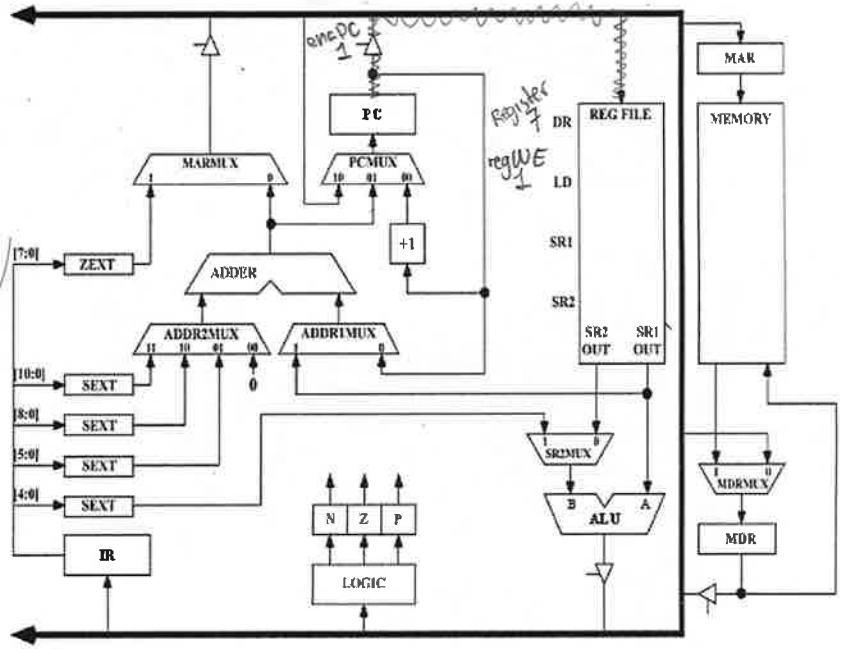
Store 1



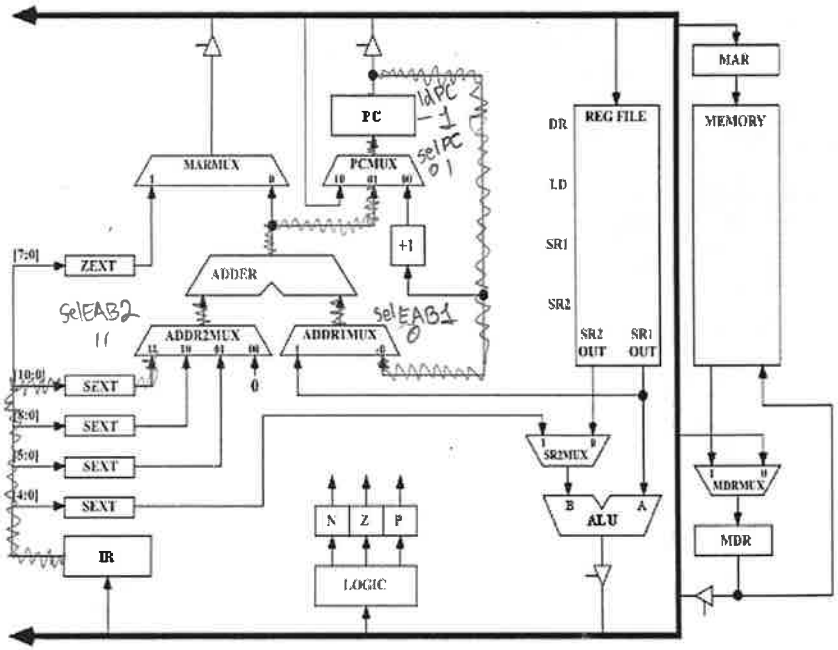
Store 2



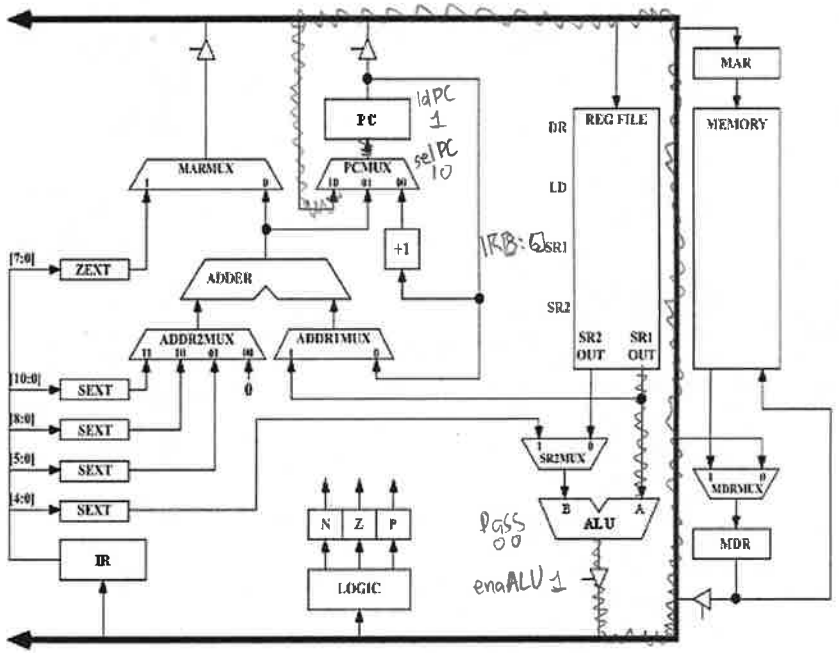
jump to Subroutine



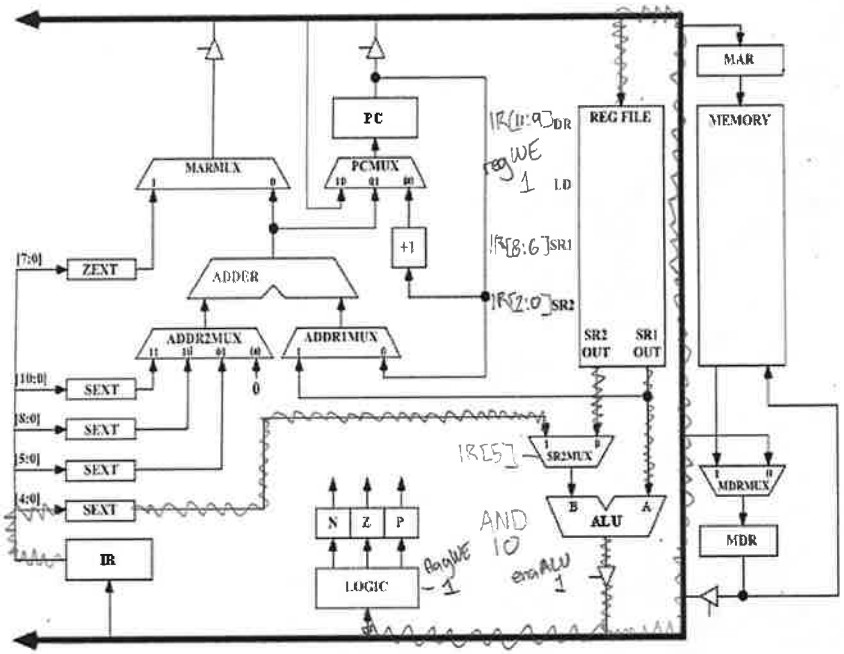
jump to Subroutine Offset [1]



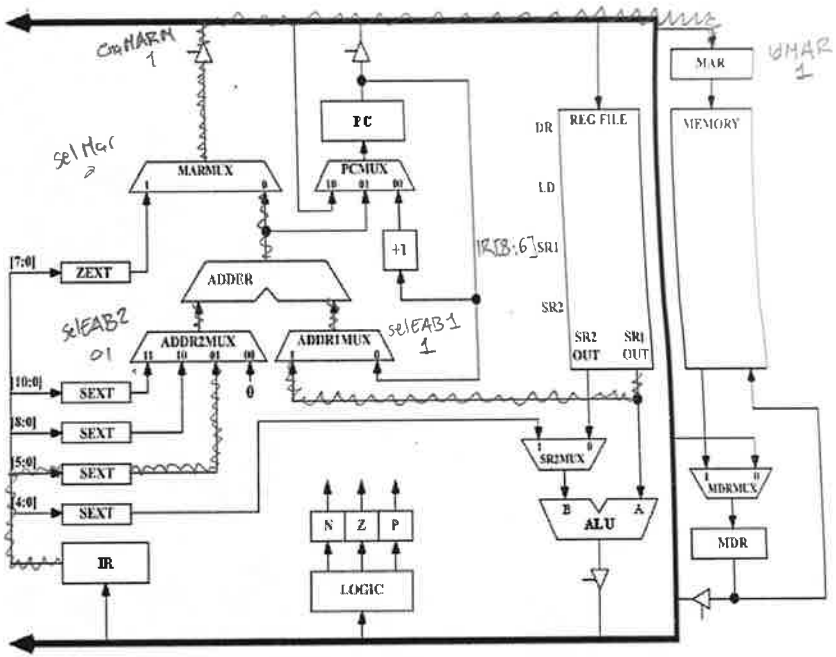
jump to Subroutine Base R [0]



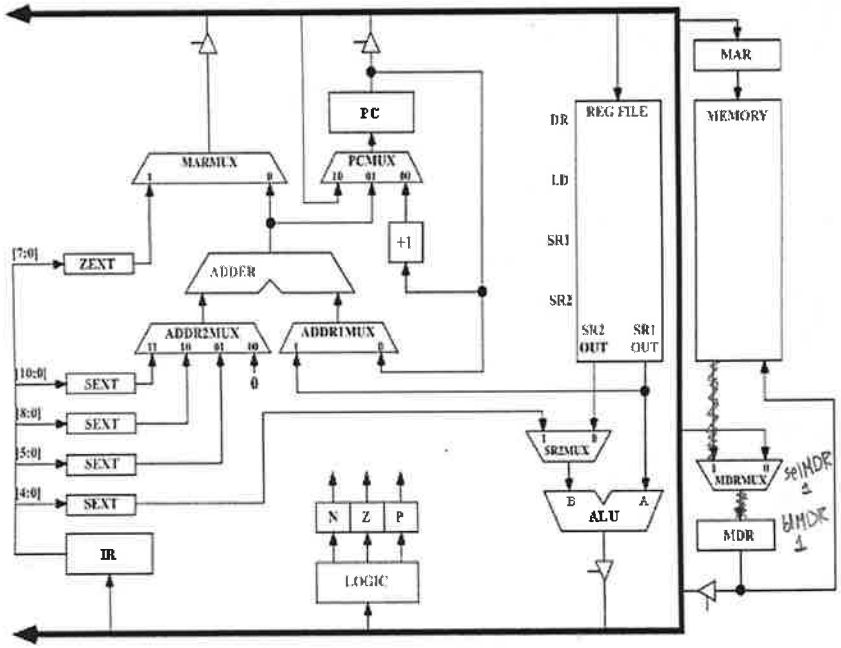
and



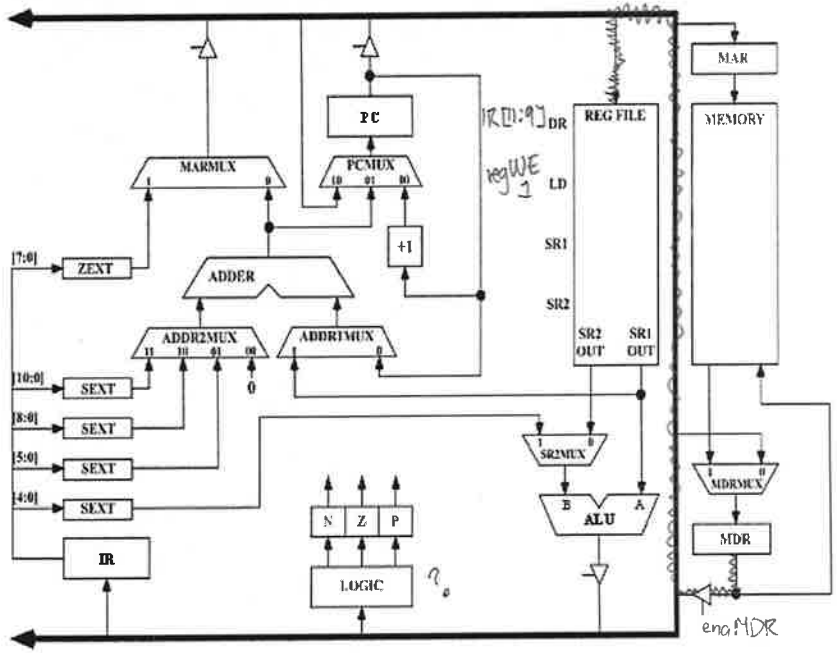
load Relative



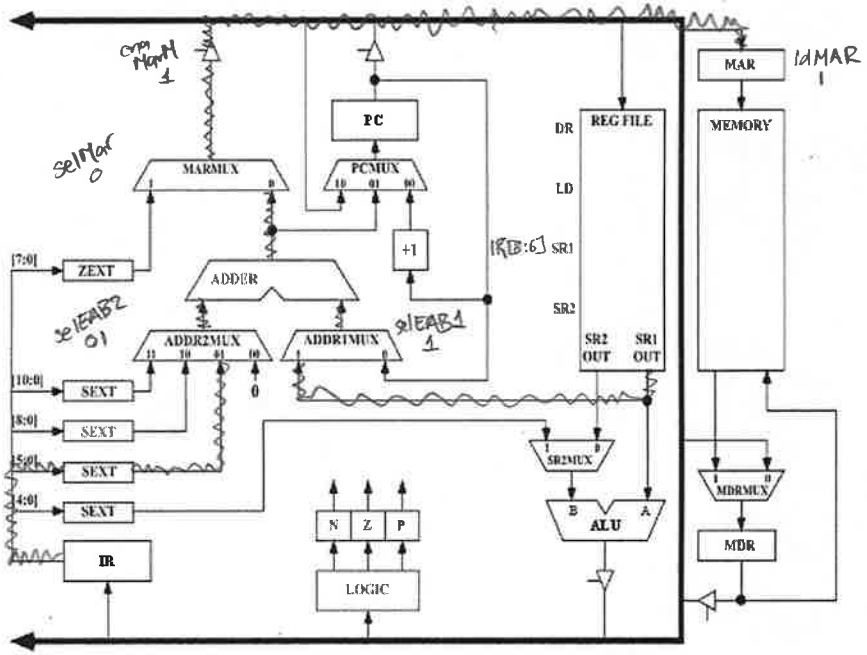
load Relative 1



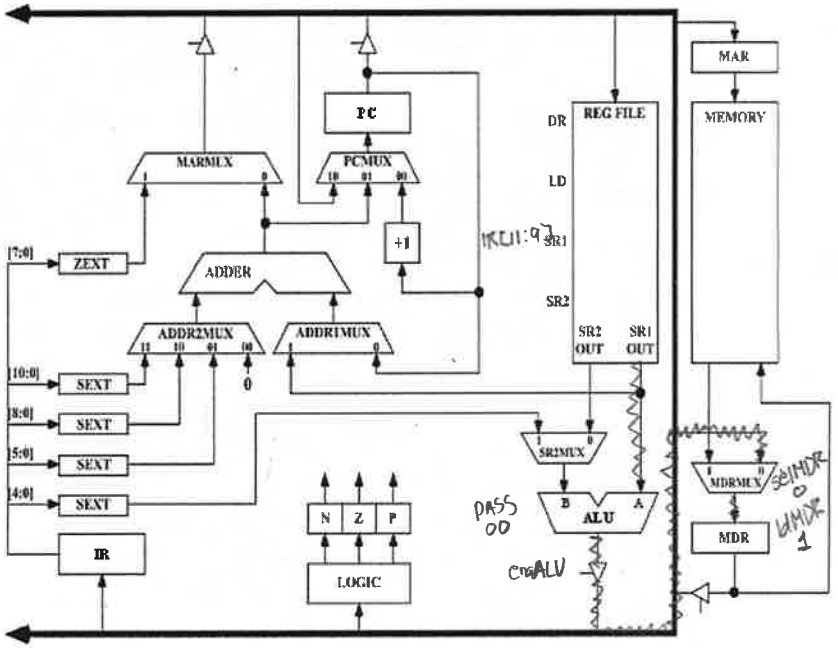
load Relative 2



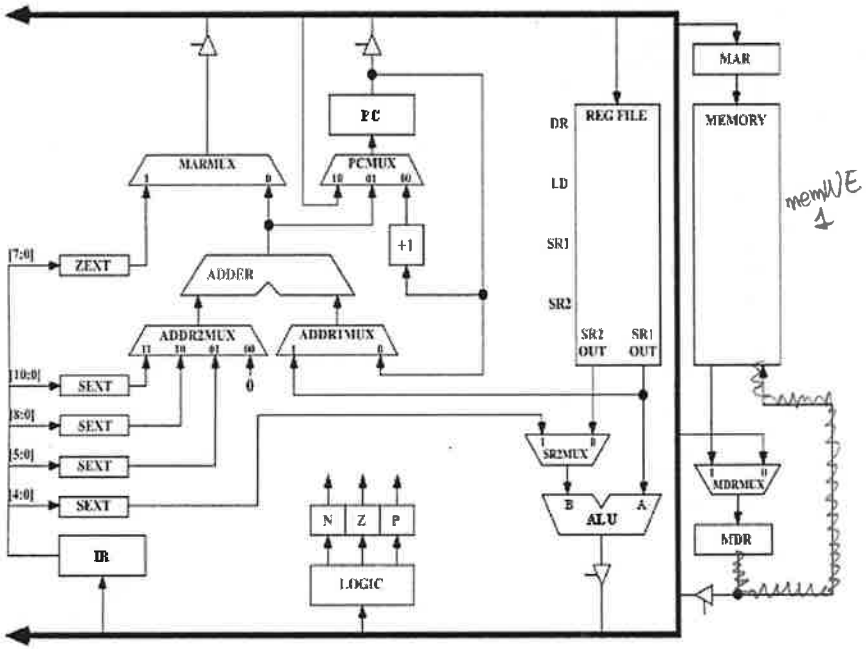
store Relative



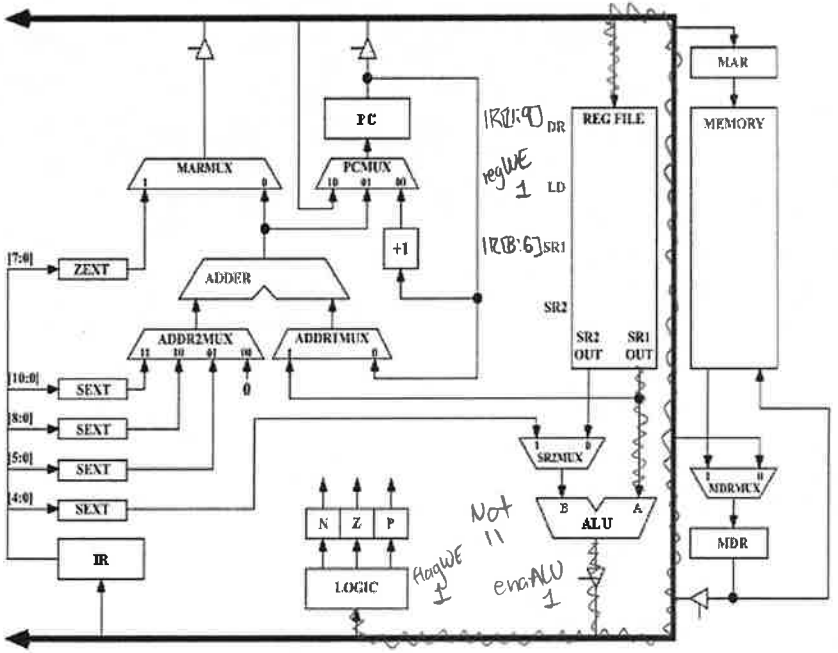
store Relative 1



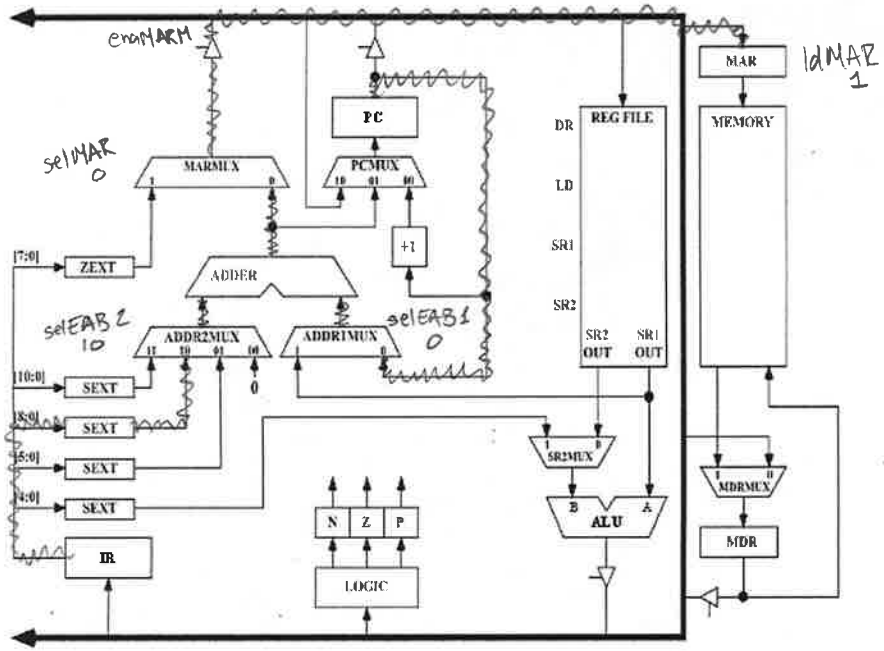
store Relative 2



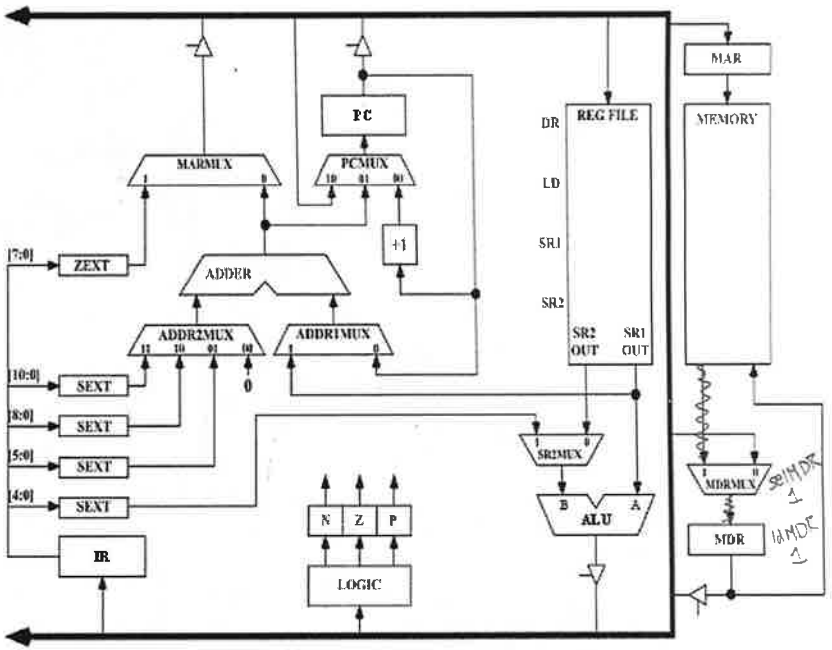
not



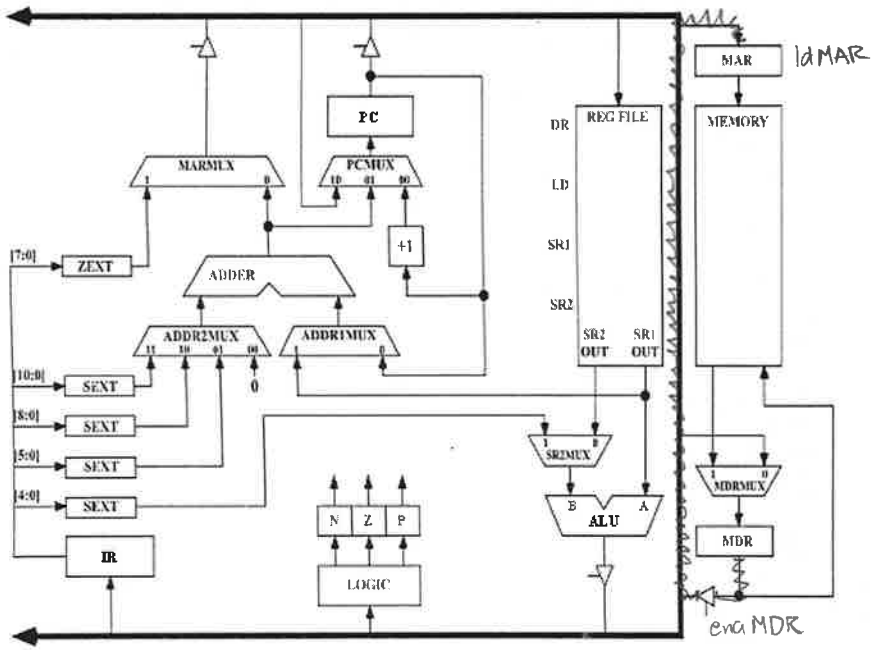
load Indirect



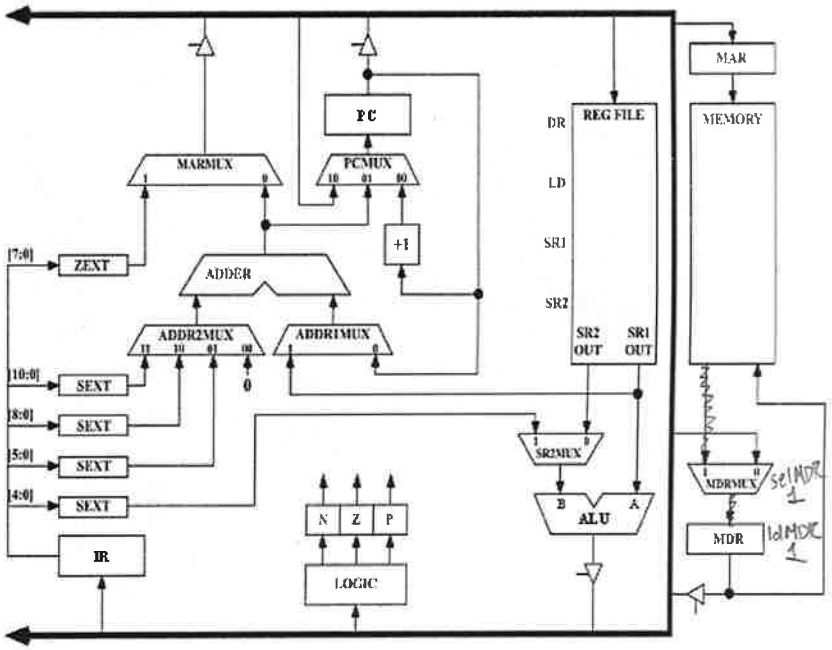
load Indirect 1



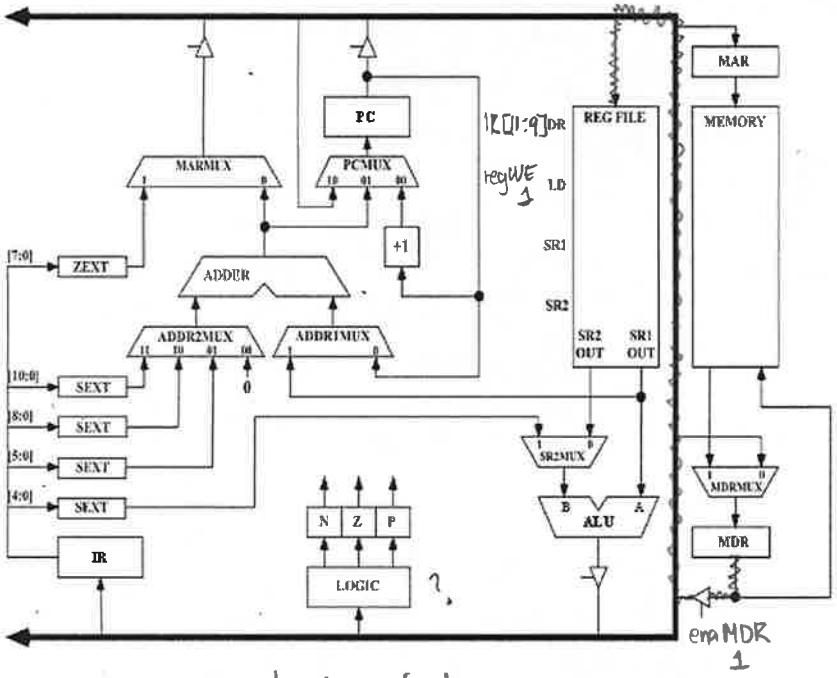
load Indirect 2



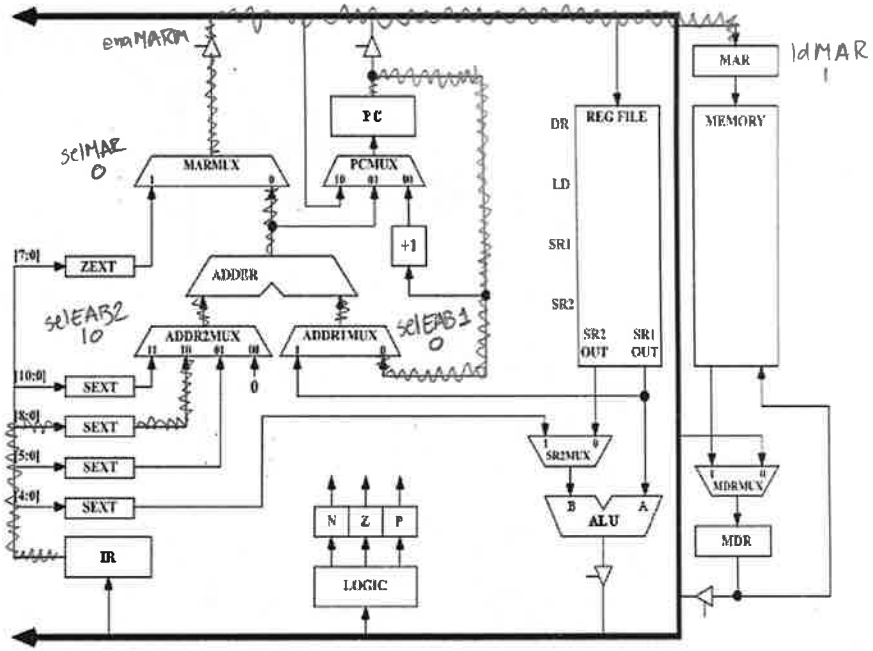
load Indirect 3



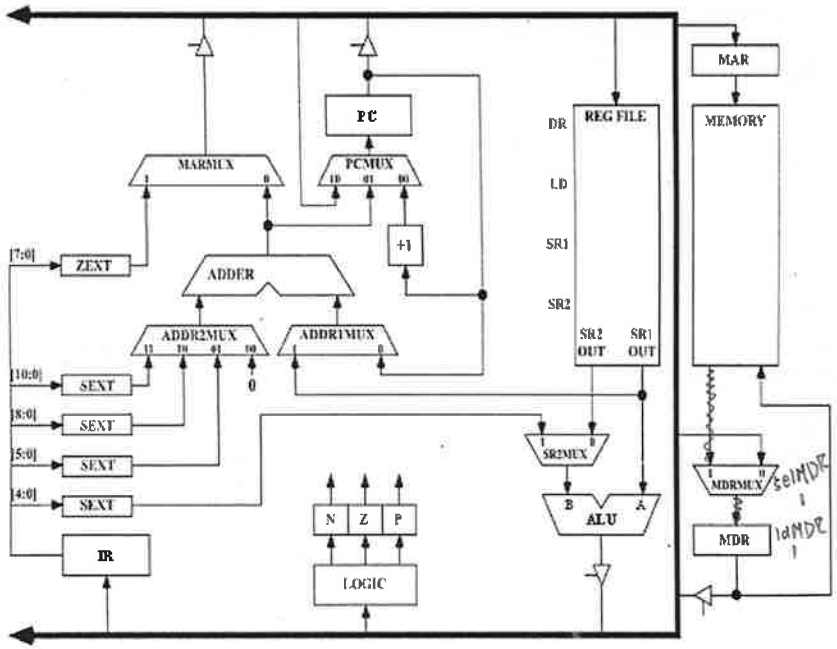
load Indirect 4



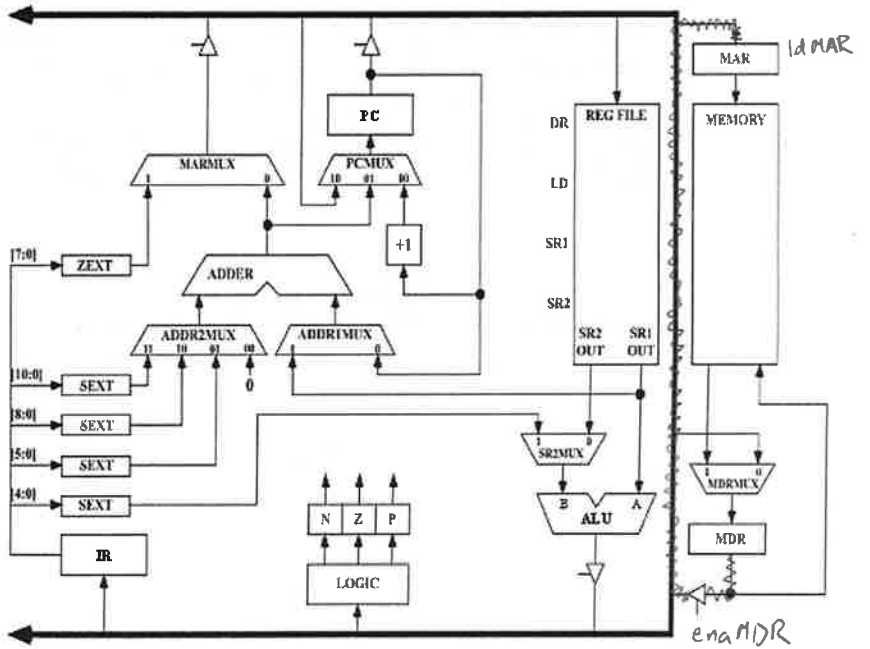
store Indirect



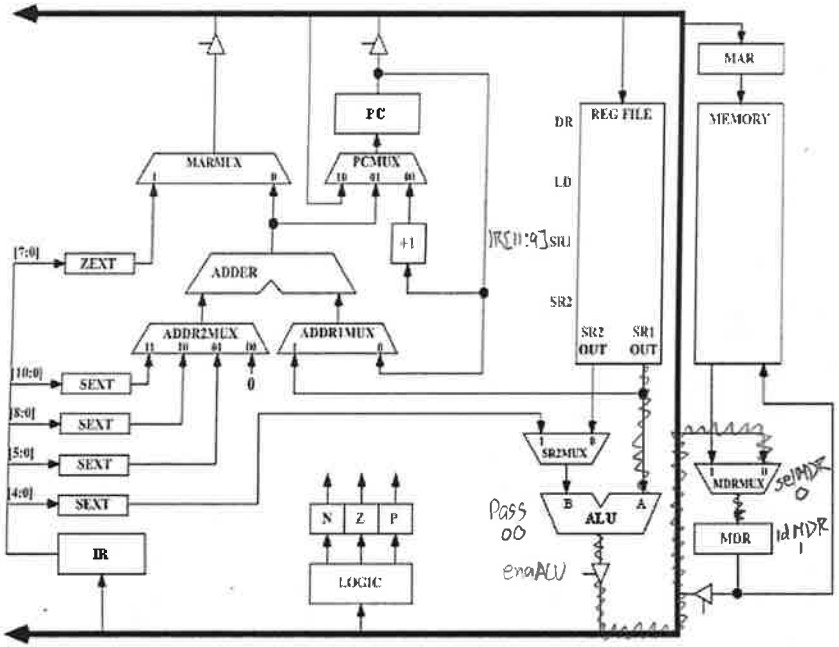
store Indirect 1



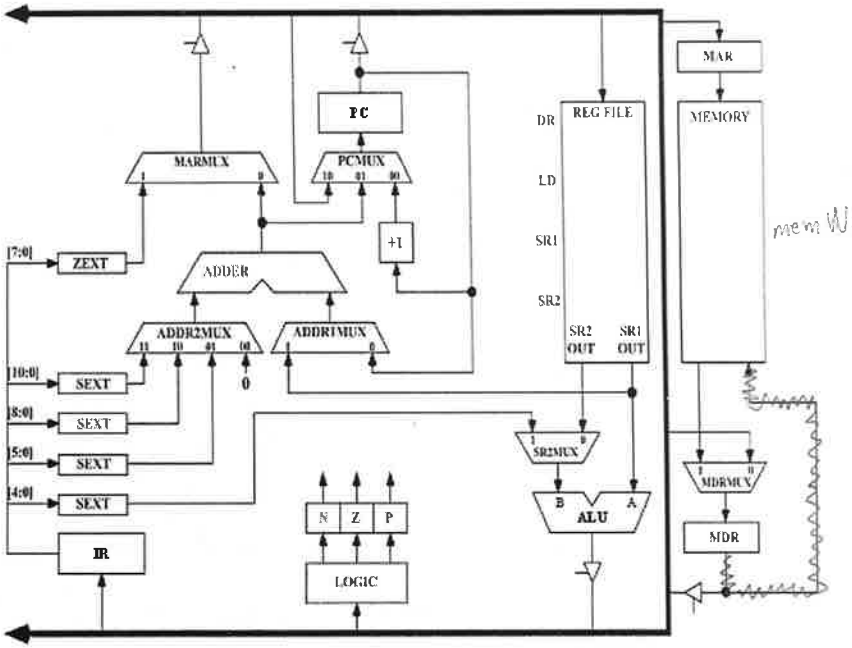
storeIndirect 2



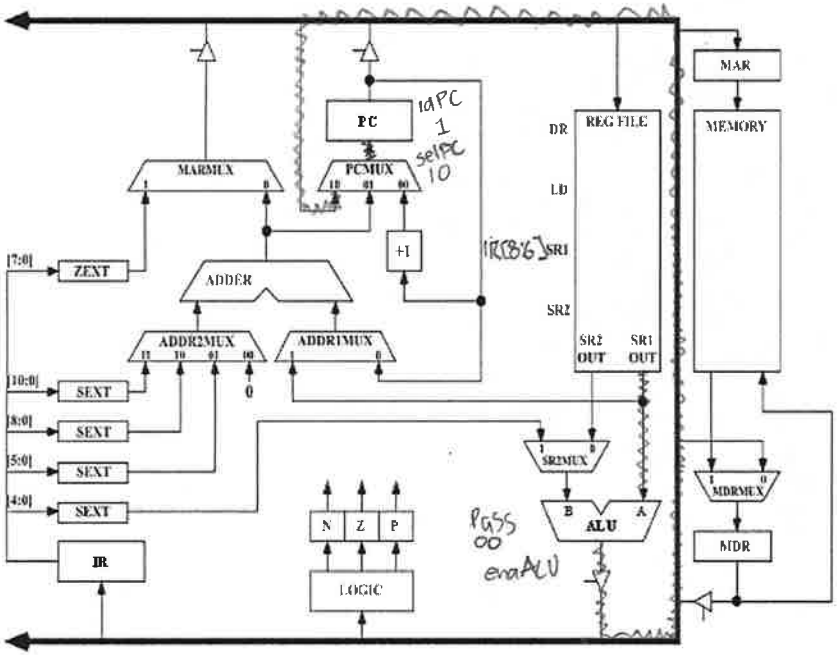
storeIndirect 3



storeIndirect 4



Jump



loadEffective Address