Bus value				Operation			Write result to			Write bus to		Condition	Jump target	
D AC IN	[D]	[X] [Y,D] [Y,X]	[Y,X++]	LOAD AND OR XOR ADD SUB	STORE	JUMP	AC	X Y	OUT	[D]	[X] [Y,D] [Y,X] [Y,X++]	AC > 0 AC = 0 AC ≥ 0 AC < 0 AC ≠ 0 AC ≤ 0	Y,D Y,[D] Y,AC Y,IN	D [D] AC IN
X	X	X	-	X	-	-	X	-	-	-	-	-	-	-
X	X	-	-	X	-	-	-	X	X	-	-	-	-	-
X	-	-	X	X	-	-	-	-	X	-	-	-	-	-
X	-	-	-	-	X	-	-	-	-	X	X	-	-	-
X	-	-	-	-	X	-	-	X*	-	X	-	-	-	-
-	X	X	X	-	X*	-	-	-	-	X	X	-	-	-
-	-	-	-	-	-	X	-	-	-	-	-	-	X	X
-	-	-	-	-	-	X	-	-	-	-	-	X	-	X

**ctrl instruction* ** AC to X/Y* *2017-04-17 Marcel van Kervinck (Last update: 2019-11-25)*

AC is the 8-bit accumulator. *X* and *Y* are 8-bit addressing registers. *IN* and *OUT* are 8-bit I/O
D is the 8-bit operand (internally cached in the Data Register)
X controls address bit 0:7, Y controls address bit 8:15
a,b composes a 16-bit address address *256a+b*
[address] is the RAM byte at the given address
Therefore *[a,b]* is 16-bit addressing and *[a]* is zero-page addressing
X++ is post-increment of *X*. Note there's no carry into *Y*
All ALU operations operate on *AC* and bus
8-bit jumps stay within the same 256-byte page, except when the jump is from *$xxFF*. In that case the jump is into the next page.
A combined memory read and store is the CTRL instruction for the I/O and RAM expander board. It writes to its control register using the address bus for data.

	Operation			Mode			Bus	
	IR7	IR6	IR5	IR4	IR3	IR2	IR1	IR0
0	LOAD			[D],AC			D	
1	AND			[X],AC			RAM	
2	OR			[Y,D],AC			AC	
3	XOR			[Y,X],AC			IN	
4	ADD			[D],X				
5	SUB			[D],Y				
6				[D],OUT				
7				[Y,X++],OUT				
0				[D]			D	
1				[X]			undef (or CTRL)	
2				[Y,D]			AC	
3				[Y,X]			IN	
4				[D],X				
5				[D],Y				
6	STORE			[D]				
7				[Y,X++]				
0				Far jump		jmp y,*bus*	D	
1				Branch	AC>0	bgt *bus*	[D]	
2					AC<0	blt *bus*	AC	
3					AC≠0	bne *bus*	IN	
4					AC=0	beq *bus*		
5					AC≥0	bge *bus*		
6					AC≤0	ble *bus*		
7	JUMP				Always	bra *bus*		