Operations				128	64	32	16	8	4	2	1	1
Immediate	imm			0	04		10	,	4		<u> </u>	
Calculate	calc			0	1							
Сору	сру			1	0							
Condition	cond			1	1							
Condition	CONG			I	· · · · · · · · · · · · · · · · · · ·							
Register 0 in	r0					0	C) (\			
	r1i					0	C					
Register 2 in	r2i					0	1					
Register 3 in	r3i					0						
	r4i					1						
Register 4 in	r5i					1						
Register 5 in	in					1						
Input	in											
Unused						1	1	1				
Deviates 0 and	-0-											
Register 0 out	r0o								0			
Register 1 out	r1o								0			
Register 2 out	r2o								0			
Register 3 out	r3o								0			
Register 4 out	r4o								1			
Register 5 out	r5o								1			
Output	out								1		0	
Unused									1	1	1	
Or	or								0			
Nand	nand								0			
Nor	nor								0		0	
And	and								0			
Add	add								1			
Sub	sub								1	0	1	
Unused									1		0	
Unused									1	1	1	
Never	never								0	0	0	
Equal to 0	jeq								0	0	1	
Less than 0	jlt								0	1	0	
Less than or Equal to 0	jle								0	1	1	
Always	jmp								1	0	0	
Not Equal to 0	jne								1	0	1	
Greater than or Equal to 0	jge								1	1	0	
Greater than 0	jgt								1	1	1	
Instruction Layouts												
-												
Сору		12	8 64	32	16	8	4	. 2	. 1			
.,						-						

		- ··								_		
		Operation		Copy from			Copy to					
									_			
Calculation		128	64		32 16	8		4	2	1		
		Operation		Unused			Condition					
						_			_			
Condition		128	64		32 16	8		4	2	1		
		Operation		Unused			Function					
Immediate		128	64		32 16	8		4	2	1		
		Operation		Value								
Info + Setup												
		"Immediate" is the term for ser		ctly to <u>reg0</u> . The	immediate value is	marked in the bit	s 1-6 of the instr	ruction byte.				
		An immediate value can range	from 0 - 63									
		Calculations always take the v		_	e function on them.	Reg1 is on the le	ft of the calculati	on				
		Example: Addition = reg1 + re	g2, Subtraction = r	eg1 - reg 2								
		Conditions always take the value in reg3 and compare it against the given condition.										
		If the condition is evaluated as true, the program counter will jump to the line of code at the value stored in reg0.										
		All conditions are tested <u>against 0</u> .										
		When programming the further	set left hit is the 12	8th hit To prog	ram a component s	imply replace low	constants with I	nigh constants and	l rewire			
		When programming, the furthest left bit is the 128th bit. To program a component, simply replace low constants with high constants and rewire. When finished, package the whole circuit as one component and copy and paste into the processor file.										
		When you have added it into the file, rewire it in place of the placeholder program component.										
Setup												
Getup												
		Toggle the DDE and CLD switches by the program component and replace the program component with your own program										
		Toggle the PRE and CLR switches by the program component and replace the program component with your own program.										
		To start executing the program, simply toggle the clock toggle switch and wait. The clock can have any allowed frequency the processor is except a few counting but as in a 250mg.										
		The clock can have any allowed frequency, the processor is capable of executing bytes in < 250ms. The "leget block" is where you get the input value. It shows both the bigger and how representation of the number.										
		The "Input block" is where you set the input value. It shows both the binary, and hex representation of the number.										
		The "Output block" is where the final result will be sent, that is if you actually program it to do so.										
		Enjoy!										
Assembly Code Syntax	Using the Replit file											
, according code cyridax	Comy are replicate											+
		Сору:	cpy (from adr) (to	adr)								
			5p) (110111 dd1) (td									
		Calculate / Condition:	calc / cond (oper	ration)								
		Sulvuidito / Sofiultion.	caic / coria (oper	audii)								
		Immediate:	imm (value)									
		rodiato.	(value)									
		Conditions are slightly tricky to	implement Reco	use the jump of	dress is always sto	red in r0. you mus	et send an immo	diate value to r0 h	efore the jump			
		Conditions are slightly tricky to	ппретнень веса	ase are Jump a	uurcoo io aiwayo SiU	ica iii io, you iilus	a senu an mille	uiale value lu 10 D	ciore ure jurrip.			

The immediate value should be the byte of code that you want to jump to, e.g. byte 3, byte 2, etc.
Your lines of code begin at byte 0, and increment by 1 for each instruction.
After the immediate instruction, you then add your jump condition, e.g. cond jne.
In the Replit program, there is a multiline input for a code block. When you have finished your code block, simply type 'end' to convert it.