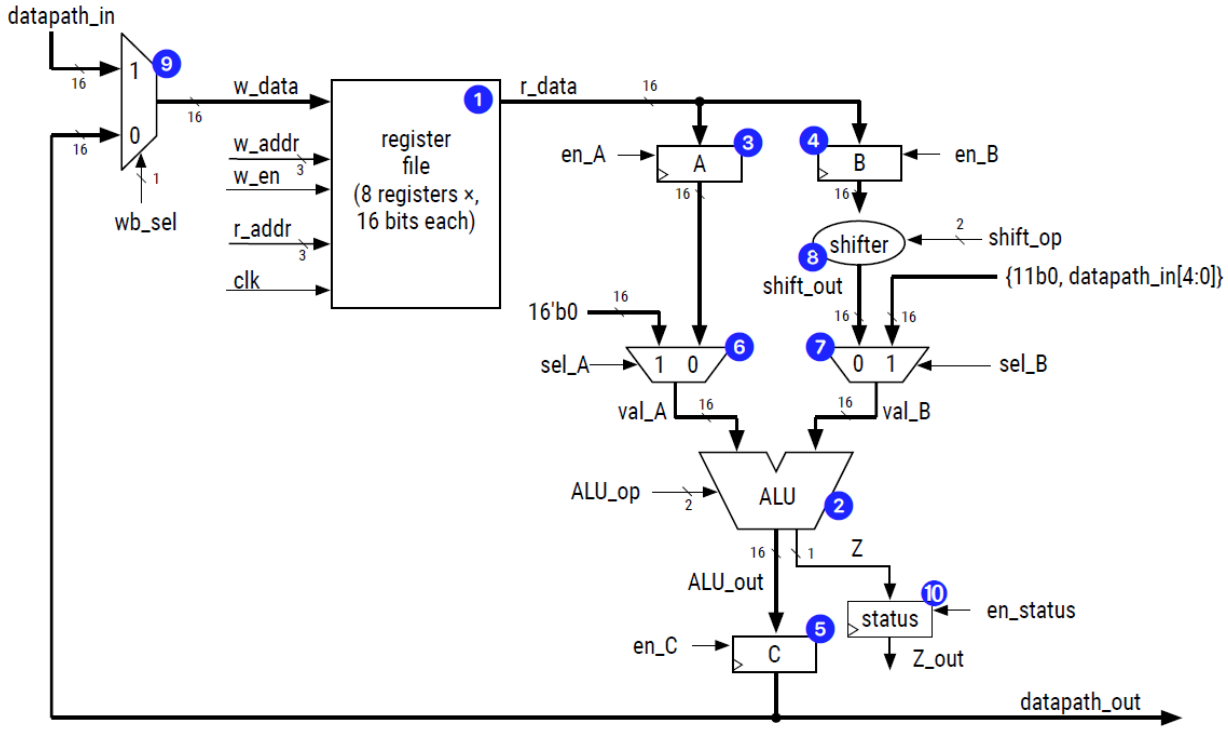
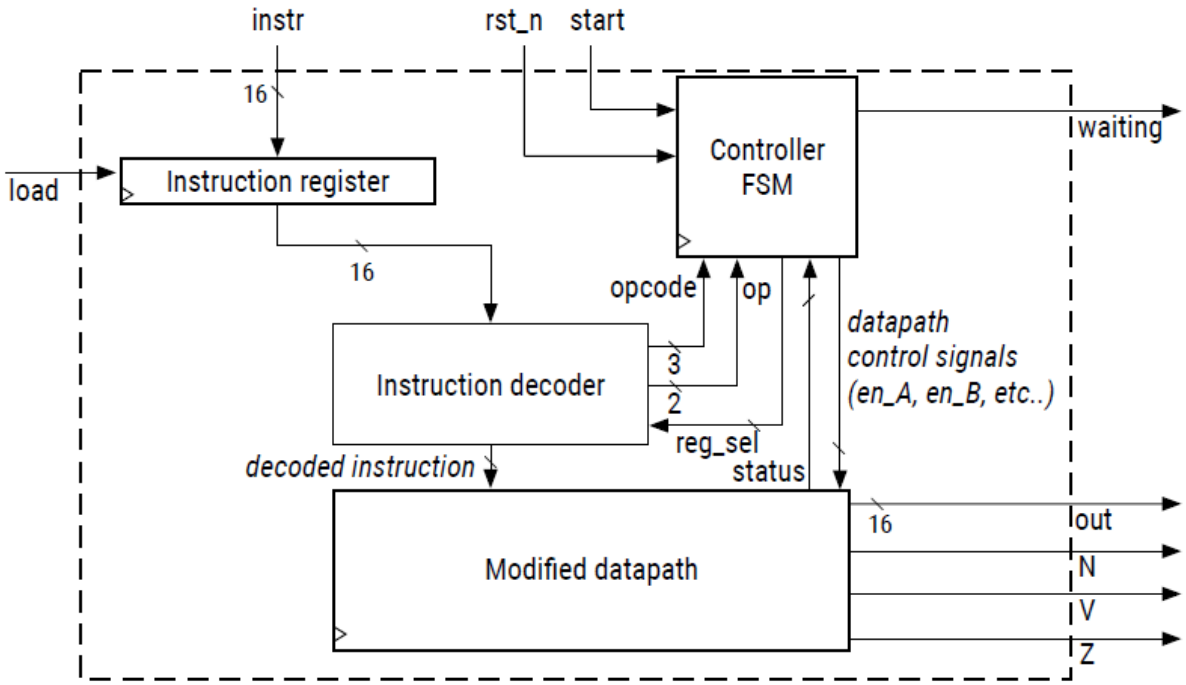
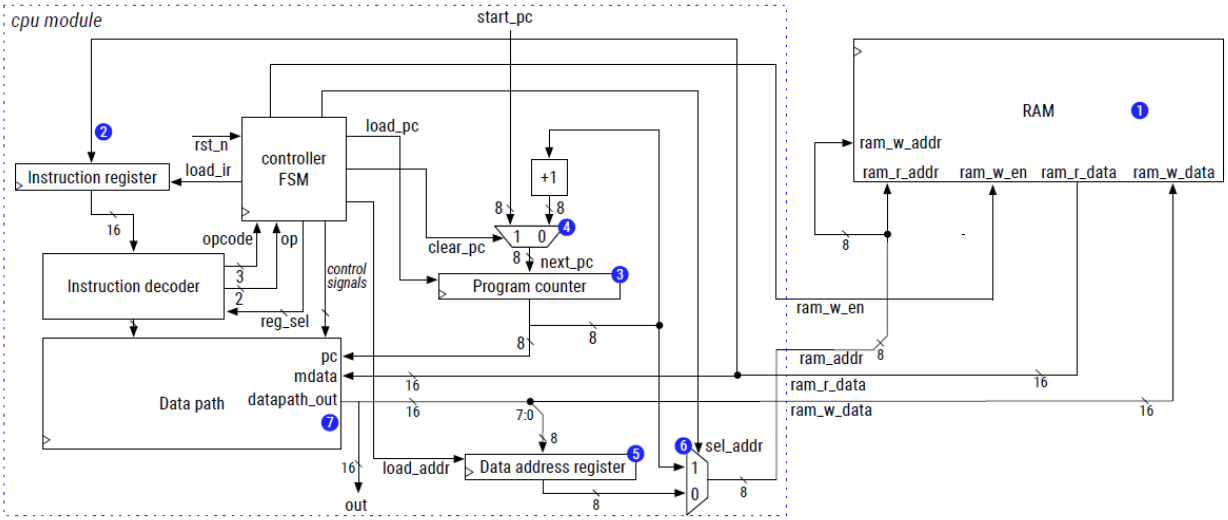
Datapath module:



CPU module:



Overall RISC machine:



Supported ISA:

Assembly syntax (see text)	Potato Machine™ 16-bit encoding																Operation (see text)
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
move instructions	*opcode*			*op*		*3b*			*8b*								
`MOV Rn,#<im8>`	1	1	0	1	0	Rn			im8								`R[Rn] = sx(im8)`
`MOV Rd,Rm{,<sh_op>}`	1	1	0	0	0	0	0	0	Rd			sh		Rm			`R[Rd] = sh_Rm`
ALU instructions	*opcode*			*ALUop*		*3b*			*3b*			*2b*		*3b*			
`ADD Rd,Rn,Rm{,<sh_op>}`	1	0	1	0	0	Rn			Rd			sh		Rm			`R[Rd]=R[Rn]+sh_Rm`
`CMP Rn,Rm{,<sh_op>}`	1	0	1	0	1	Rn			0	0	0	sh		Rm			`status=f(R[Rn]-sh_Rm)`
`AND Rd,Rn,Rm{,<sh_op>}`	1	0	1	1	0	Rn			Rd			sh		Rm			`R[Rd]=R[Rn]&sh_Rm`
`MVN Rd,Rm{,<sh_op>}`	1	0	1	1	1	0	0	0	Rd			sh		Rm			`R[Rd]=~sh_Rm`
memory instructions	*opcode*			*ALUop*		*3b*			*3b*			*5b*					
`LDR Rd,[Rn{,#<im5>}]`	0	1	1	0	0	Rn			Rd			im5					`R[Rd]=M[R[Rn]+sx(im5)]`
`STR Rd,[Rn{,#<im5>}]`	1	0	0	0	0	Rn			Rd			im5					`M[R[Rn]+sx(im5)]=R[Rd]`