INSTRUCCIONES BASADAS EN RISC-V

Instruction	Format	Description	Why it will be Implemented
DATA MANAGEMENT			
MULV	MULV vd, vs1, vs2	Multiplies corresponding elements of two vector registers and stores the result in a destination vector register.	Needed for convolution operation
ADDV	ADDV vd, vs1, vs2	Adds corresponding elements of two vector registers and stores the result in a destination vector register.	Needed for prefix sum, which is needed for convolution
LDE	LDE rd, rs1, imm	Loads a scalar value from memory into a scalar register.	Basic CPU functionality
STE	STE rd, rs1, imm	Stores a scalar value from a scalar register into memory.	Basic CPU functionality
JMP	JMP label	Unconditional jump to a target address.	Needed for the algorithm to be implemented
CMP	CMP rs1, rs2	Compares two scalar registers and sets condition flags accordingly.	Needed for JMP operation to function properly as needed
ROTV	ROTV vd, vs1, imm	Rotates elements in a vector register by a specified immediate amount.	Needed for prefix sum, which is needed for convolution.
MEMORY ACCESS			
LDV	LDV vd, rs1, imm	Loads a vector of data elements from memory into a vector register.	Basic CPU functionality
STV	STV vd, rs1, imm	Stores a vector of data elements from a vector register into memory.	Basic CPU functionality