Custom Processor — Instruction Set (24-bit ISA)

Instruction size:	24 bits (H5..H0, each nibble = 4 bits)
Registers:	16 general-purpose registers R0..R15, 16-bit width
Address width:	16 bits
Register placement:	RD = H3, RS1 = H2, RS2 = H1, RD2 = H0 (when used)
Flags:	C (Carry), Z (Zero), S (Sign). Compare instructions set EQ, LT, GT flags.

LOAD_IMM (H5 = 0000)

H5	H4	H3	H2	H1	H0
0000	IMM(15:12)	RD	IMM(11:8)	IMM(7:4)	IMM(3:0)

Operation: RD ← IMM(15:0)
Description: Loads a 16-bit immediate value directly into RD.

LOAD_ABS (H5 = 0001)

H5	H4	H3	H2	H1	H0
0001	IMM(15:12)	RD	IMM(11:8)	IMM(7:4)	IMM(3:0)

Operation: RD ← MEM[IMM(15:0)]
Description: Load 16-bit word from absolute 16-bit address into RD.

LOAD_BASE_OFFSET (H5 = 0010)

H5	H4	H3	H2	H1	H0
0010	IMM(11:8)	RD	RS1	IMM(7:4)	IMM(3:0)

Operation: RD ← MEM[RS1 + IMM(11:0)]
Description: Loads data from the address obtained by adding RS1 and a 12-bit immediate offset.

STORE_BASE_OFFSET (H5 = 0011)

H5	H4	H3	H2	H1	H0
0011	IMM(11:8)	IMM(7:4)	RS1	RS2	IMM(3:0)

Operation: MEM[RS1 + IMM(11:0)] ← RS2
Description: Stores RS2 into memory at the address formed by RS1 + 12-bit immediate.

Arithmetic group (H5=0100)

MOV (H4=0000)

H5	H4	H3	H2	H1	H0
0100	0000	RD	RS1	0000	0000

Operation: RD ← RS1
Description: Moves RS1 into RD.

NEG (H4=0001)

H5	H4	H3	H2	H1	H0
0100	0001	RD	RS1	0000	0000

Operation: RD ← ~RS1 + 1
Description: Two's complement negation of RS1.

ADD (H4=0010)

H5	H4	H3	H2	H1	H0
0100	0010	RD	RS1	RS2	0000

Operation: RD ← RS1 + RS2
Description: Adds RS1 and RS2.

SUB (H4=0011)

H5	H4	H3	H2	H1	H0
0100	0011	RD	RS1	RS2	0000

Operation: RD ← RS1 + ~RS2 + 1
Description: Subtracts RS2 from RS1.

MUL (H4=0100)

H5	H4	H3	H2	H1	H0
0100	0100	RD1	RS1	RS2	RD2

Operation: {RD1(MSB), RD2(LSB)} ← RS1 × RS2
Description: Unsigned 16×16 multiply; MSB→RD1, LSB→RD2.

SMUL (H4=0101)

H5	H4	H3	H2	H1	H0
0100	0101	RD1	RS1	RS2	RD2

Operation: {RD1, RD2} ← signed(RS1) × signed(RS2)
Description: Signed multiply; MSB→RD1, LSB→RD2.

UDIV (H4=0110) - Reserved

H5	H4	H3	H2	H1	H0
0100	0110	RD	RS1	RS2	0000

Operation: Reserved
Description: Reserved for unsigned division.

SDIV (H4=0111) - Reserved

H5	H4	H3	H2	H1	H0
0100	0111	RD	RS1	RS2	0000

Operation: Reserved
Description: Reserved for signed division.

CMPU (H4=1000)

H5	H4	H3	H2	H1	H0
0100	1000	0000	RS1	RS2	0000

Operation: Compare_unsigned(RS1, RS2)
Description: Sets EQ/LT/GT flags (unsigned).

CMPS (H4=1001)

H5	H4	H3	H2	H1	H0
0100	1001	0000	RS1	RS2	0000

Operation: Compare_signed(RS1, RS2)
Description: Sets EQ/LT/GT flags (signed).

ROL (H4=1010)

H5	H4	H3	H2	H1	H0
0100	1010	RD	RS1	IMM	0000

Operation: RD ← RotateLeft(RS1, IMM)
Description: Rotate RS1 left by IMM (0–15).

ROR (H4=1011)

H5	H4	H3	H2	H1	H0
0100	1011	RD	RS1	IMM	0000

Operation: RD ← RotateRight(RS1, IMM)
Description: Rotate RS1 right by IMM (0–15).

CMPUI (H4=1100)

H5	H4	H3	H2	H1	H0
0100	1100	IMM(11:8)	RS1	IMM(7:4)	IMM(3:0)

Operation: Compare_unsigned(RS1, IMM(11:0))
Description: Unsigned compare with 12-bit immediate.

SHL (H4=1101)

H5	H4	H3	H2	H1	H0
0100	1101	RD	RS1	IMM	0000

Operation: RD ← RS1 << IMM
Description: Logical left shift.

SHRL (H4=1110)

H5	H4	H3	H2	H1	H0
0100	1110	RD	RS1	IMM	0000

Operation: RD ← RS1 >> IMM (logical)
Description: Logical right shift.

SHRA (H4=1111)

H5	H4	H3	H2	H1	H0
0100	1111	RD	RS1	IMM	0000

Operation: RD ← ArithmeticShiftRight(RS1, IMM)
Description: Arithmetic right shift.

Logic group (H5=1000)

NOT (H4=0000)

H5	H4	H3	H2	H1	H0
1000	0000	RD	RS1	0000	0000

Operation: RD ← ~RS1
Description: Bitwise NOT.

OR (H4=0001)

H5	H4	H3	H2	H1	H0
1000	0001	RD	RS1	RS2	0000

Operation: RD ← RS1 | RS2
Description: Bitwise OR.

AND (H4=0010)

H5	H4	H3	H2	H1	H0
1000	0010	RD	RS1	RS2	0000

Operation: RD ← RS1 & RS2
Description: Bitwise AND.

XOR (H4=0011)

H5	H4	H3	H2	H1	H0
1000	0011	RD	RS1	RS2	0000

Operation: RD ← RS1 ^ RS2
Description: Bitwise XOR.

DOUBLE NOT (H4=0100)

H5	H4	H3	H2	H1	H0
1000	0100	RD	RS1	0000	0000

Operation: RD ← ~~RS1
Description: Double NOT.

NOR (H4=0101)

H5	H4	H3	H2	H1	H0
1000	0101	RD	RS1	RS2	0000

Operation: RD ← ~(RS1 | RS2)
Description: Bitwise NOR.

NAND (H4=0110)

H5	H4	H3	H2	H1	H0
1000	0110	RD	RS1	RS2	0000

Operation: RD ← ~(RS1 & RS2)
Description: Bitwise NAND.

XNOR (H4=0111)

H5	H4	H3	H2	H1	H0
1000	0111	RD	RS1	RS2	0000

Operation: RD ← ~(RS1 ^ RS2)
Description: Bitwise XNOR.

Branch group (H5=1100)

JEZ (H4=0000)

H5	H4	H3	H2	H1	H0
1100	0000	ADDR(15:12)	ADDR(11:8)	ADDR(7:4)	ADDR(3:0)

Operation: Jump if EQ = 1
Description: Branch to ADDR when EQ flag set.

JCS (H4=0001)

H5	H4	H3	H2	H1	H0
1100	0001	ADDR(15:12)	ADDR(11:8)	ADDR(7:4)	ADDR(3:0)

Operation: Jump if C = 1
Description: Branch when carry set.

JNS (H4=0010)

H5	H4	H3	H2	H1	H0
1100	0010	ADDR(15:12)	ADDR(11:8)	ADDR(7:4)	ADDR(3:0)

Operation: Jump if S = 1
Description: Branch when sign set (negative).

Reserved (H4=0011)

H5	H4	H3	H2	H1	H0
1100	0011	ADDR(15:12)	ADDR(11:8)	ADDR(7:4)	ADDR(3:0)

Operation: Reserved
Description: Reserved.

JNEZ (H4=0100)

H5	H4	H3	H2	H1	H0
1100	0100	ADDR(15:12)	ADDR(11:8)	ADDR(7:4)	ADDR(3:0)

Operation: Jump if EQ = 0
Description: Branch when not equal.

JNC (H4=0101)

H5	H4	H3	H2	H1	H0
1100	0101	ADDR(15:12)	ADDR(11:8)	ADDR(7:4)	ADDR(3:0)

Operation: Jump if C = 0
Description: Branch when carry clear.

JPS (H4=0110)

H5	H4	H3	H2	H1	H0
1100	0110	ADDR(15:12)	ADDR(11:8)	ADDR(7:4)	ADDR(3:0)

Operation: Jump if S = 0
Description: Branch when sign clear (positive).

Reserved (H4=0111)

H5	H4	H3	H2	H1	H0
1100	0111	ADDR(15:12)	ADDR(11:8)	ADDR(7:4)	ADDR(3:0)

Operation: Reserved
Description: Reserved.

JMP (H4=1000)

H5	H4	H3	H2	H1	H0
1100	1000	ADDR(15:12)	ADDR(11:8)	ADDR(7:4)	ADDR(3:0)

Operation: Unconditional jump
Description: Always branch to ADDR.

JEQ (H4=1001)

H5	H4	H3	H2	H1	H0
1100	1001	ADDR(15:12)	ADDR(11:8)	ADDR(7:4)	ADDR(3:0)

Operation: Jump if EQ = 1
Description: Branch when equal.

JGT (H4=1010)

H5	H4	H3	H2	H1	H0
1100	1010	ADDR(15:12)	ADDR(11:8)	ADDR(7:4)	ADDR(3:0)

Operation: Jump if GT = 1
Description: Branch when greater.

JLT (H4=1011)

H5	H4	H3	H2	H1	H0
1100	1011	ADDR(15:12)	ADDR(11:8)	ADDR(7:4)	ADDR(3:0)

Operation: Jump if LT = 1
Description: Branch when less.

Reserved (H4=1100)

H5	H4	H3	H2	H1	H0
1100	1100	ADDR(15:12)	ADDR(11:8)	ADDR(7:4)	ADDR(3:0)

Operation: Reserved
Description: Reserved.

JNE (H4=1101)

H5	H4	H3	H2	H1	H0
1100	1101	ADDR(15:12)	ADDR(11:8)	ADDR(7:4)	ADDR(3:0)

Operation: Jump if EQ = 0
Description: Branch when not equal.

JLE (H4=1110)

H5	H4	H3	H2	H1	H0
1100	1110	ADDR(15:12)	ADDR(11:8)	ADDR(7:4)	ADDR(3:0)

Operation: Jump if LT=1 OR EQ=1
Description: Branch when less or equal.

JGE (H4=1111)

H5	H4	H3	H2	H1	H0
1100	1111	ADDR(15:12)	ADDR(11:8)	ADDR(7:4)	ADDR(3:0)

Operation: Jump if GT=1 OR EQ=1
Description: Branch when greater or equal.

Flags and notes

C (Carry):	Set/cleared by arithmetic; used by JCS/JNC.
Z (Zero):	Generic zero flag; EQ flag is used for compare results.
S (Sign):	Sign/MSB of result; used by JNS/JPS.
EQ/LT/GT:	Dedicated compare-result flags set by CMPU/CMPS; used by JEQ/JNE/JGT/JLT/JLE/JGE.
CMPU/CMPS:	CMPU = unsigned compare; CMPS = signed two's-complement compare.