Hardware Specification

1. **Arithmetic Logic Unit (ALU):**
   The DLW-1 contains a simple ALU with two operations, addition and subtraction.

2. **General Purpose Registers (GPRs):**
   The DLW-1 contains four 8-bit (1 byte) general-purpose registers, identified A, B, C, and D.

3. **Processor Status Word (PSW):**
   The DLW-1 has a 2-bit processor status word register, used to store condition flags for branching based on the previous arithmetic operation. The bit layout is:

1	**0**
Negative	Zero

4. **Program Counter (PC):**
   The PC is an 8-bit (1 byte) register that stores the address of the current instruction in memory.

5. **Instruction Register (IR):**
   The IR is a 16-bit (2 byte) register that stores the current instruction being executed.

6. **Memory Model:**
   The DLW-1 uses an 8-bit address space, allowing only direct access of 256 bytes in a single memory bank. To increase the total memory, the 8-bit Current Bank Register (CBR) specifies which 256-byte memory bank is currently active, giving up to 65536 bytes of total accessible memory. All instructions using memory addresses operate relative to the currently selected bank.

Behavior

1. **Signed vs. Unsigned Interpretation:**
   GPRs and memory store raw 8-bit values (0-255). For arithmetic and conditional branching, values are interpreted as signed two's complement integers (-128 to 127). Immediate values in instructions are also treated as signed when used for relative addressing.

2. **Fetch-Decode-Execute Model:**
   Fetch: The instruction at the address in PC is read from memory (in the current bank) into the IR. PC is incremented by 2 (wraps to 0 after 255).
   Decode: The binary instruction in IR is translated into an internal representation of its opcode, addressing mode, registers, and immediates.
   Execute: The instruction is performed, affecting PC, PSW, GPRs, CBR, and/or memory as required.

3. **Boot:**
   On reset, PC, PSW, IR, CBR, and all GPRs are cleared to 0. All memory banks are cleared. The loader places the binary program into memory starting at address 0 of bank 0. If the program file exceeds 256 bytes, it is split across sequential banks by the loader. At runtime, the CPU will only access the current bank unless explicitly switched via a bank-switch instruction.

4. **Halt:**
   The program runs until a HALT instruction is executed or PC wraparound occurs.

5. **PSW Behavior:**
   The PSW is updated only by arithmetic instructions. Conditional branches (jumps) read but do not write PSW flags.

Instruction Set

Below are the binary representations of DLW-1 instructions:

Instruction	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADD (immediate)	imm								dest		src		000			1
ADD (register)	000000						dest		src2		src		000			0
SUB (immediate)	imm								dest		src		001			1
SUB (register)	000000						dest		src2		src		001			0
LOAD (immediate)	imm								dest		00		010			1
LOAD (register)	000000						dest		00		src		010			0
LOAD (relative)	imm								dest		src		010			1
BANK SWITCH	imm								1111				010			0
STORE (immediate)	imm								src		00		011			1
STORE (register)	000000						dest		00		src		011			0
STORE (relative)	imm								src2		src		011			1
MOV	000000						dest		11		src		011			0
JUMP (immediate)	imm								0000				100			1
JUMP (register)	0000000000										src		100			0
JUMP (relative)	imm									00		1	100			1
HALT	11111111								0000				100			0
JUMPZ (immediate)	imm								0000				101			1
JUMPZ (register)	0000000000										src		101			0
JUMPZ (relative)	imm									00		1	101			1
JUMPNZ (immediate)	imm								0000				110			1
JUMPNZ (register)	0000000000										src		110			0
JUMPNZ (relative)	imm									00		1	110			1
JUMPN (immediate)	imm								0000				111			1
JUMPN (register)	0000000000										src		111			0
JUMPN (relative)	imm									00		1	111			1

Notes

1. **Immediates:**
   - Arithmetic instructions (ADD, SUB): The immediate field is treated as an 8-bit signed operand (two's complement) to be used directly in the ALU operation.
   - Memory instructions (LOAD, STORE): The immediate field is an 8-bit unsigned address within the current memory bank.

   - BANK SWITCH: The immediate field specifies the bank number (0–255) to select in the CBR.
   - Branch instructions (JUMP, JUMPZ, JUMPNZ, JUMPN):
     - Immediate form: The immediate is the absolute address (0–255) in the current bank to jump to.
     - Relative form: The immediate is a signed 9-bit offset (-256 to +255) added to the PC after it has been incremented. Wrap-around is allowed.

2. **Registers:**
   - Arithmetic instructions: Specify source (operands) and destination (sum/difference) GPRs.
   - Memory instructions:
     - Contains either the base address register or the data register depending on the operation:
       - LOAD: source is the memory address (from a register), destination is the GPR to receive the data.
       - STORE: source is the GPR containing the data, destination is the register holding the target memory address.
     - In relative forms, one register holds a base address, and the immediate offset is added (signed) before accessing memory.
   - Branch instructions: Holds the absolute address within the current bank to jump to.

Assembler

Coming Soon!