1. Memory Structures for the GBVM Virtual Machine

File: `sasm_memory.h`
Author: Soham Metha
Date: January 2025

This file contains the declaration of memory structures utilized by the GBVM (GameBoy Virtual Machine).

1.1. Table of Contents

1.2. Structures

1.2.1. Registers

Description:
Represents the general-purpose registers of the CPU, along with the stack pointer (SP) and instruction pointer (IP).

Members:

Member	Type	Description
`AX`	`Word`	Accumulator register.
`BX`	`Word`	Base register.
`CX`	`Word`	Counter register.
`DX`	`Word`	Data register.
`SP`	`Word`	Stack pointer register.
`IP`	`Word`	Instruction pointer register.

1.2.2. CPU

Description:
Represents the central processing unit (CPU) of the virtual machine, which includes its registers and flags.

Members:

Member	Type	Description
`registers`	`Registers`	The CPU's general registers.
`flags`	`Word`	CPU flags for condition checking.

1.2.3. Memory

Description:
Represents the system's memory, primarily focused on stack memory.

Members:

Member	Type	Description
`stack`	`Word[]`	Stack memory, defined by `STACK_CAPACITY`.

1.3. Example Usage

1.3.1. Initializing the CPU

```
#include "sasm_memory.h"

int main() {
    CPU cpu = {{0, 0, 0, 0, 0, 0}, 0};  // Initialize all registers and
flags to zero.
    cpu.registers.AX = 10;             // Set accumulator register.
    cpu.flags = 1;                     // Set a flag value.
    return 0;
}
```

1.3.2. Working with Memory

```
#include "sasm_memory.h"

int main() {
    Memory memory;
    memory.stack[0] = 42;  // Push a value onto the stack.
    printf("Stack[0]: %d\n", memory.stack[0]);  // Output: Stack[0]: 42
    return 0;
}
```