MemoryCom Class Documentation

Source File: ‘`Component.h`’
Namespace: ca
Class Header: `class MemoryCom : public Executable, public Object`

Overview

The *MemoryCom* abstract class is a template for computer memory.

Constructors

- `MemoryCom()` (default constructor)
  - **Purpose**: Constructs empty memory storage without linking a bus.
- `MemoryCom(const MemoryCom& obj)` (copy constructor)
  - **Purpose**: Performs a deep copy of *obj* without linking a bus.
  - **Parameter(s)**:
    - *obj*: Constant *MemoryCom* reference object.
- `MemoryCom(size_t sz)`
  - **Purpose**: Constructs memory storage of size *sz* such that each word is 32-bit without linking a bus if *sz* is greater than 0; otherwise, an empty memory.
  - **Parameter(s)**:
    - *sz*: Number of storage locations.
- `MemoryCom(size_t sz,size_t bt)`
  - **Purpose**: Constructs memory storage of size *sz* such that each word is *bt*-bit without linking a bus if *sz* is greater than 0; otherwise, an empty. If *bt* is 0, each word will be 32 bits.
  - **Parameter(s)**:
    - *sz*: Number of storage locations.
    - *bt*: Size of words.

Destructor

- `~MemoryCom()` [virtual]
  - **Purpose**: Deallocates memory storage and unlinks the bus.

Assignment Operator

- `operator=(const MemoryCom& rhs)`
  - **Purpose**: Performs a deep copy of *rhs* without linking a bus.
  - **Parameter(s)**:
    - *rhs*: Constant *MemoryCom* reference object.
  - **Return**: `*this`.

Methods

- `locations() const`
  - **Purpose**: Gets the number of locations in the memory.
  - **Return**: The number of memory locations.
- `bits() const`
  - **Purpose**: Gets the length of the words in the memory.
  - **Return**: The number of bits of a location.
- `link(BusCom& bus)` [overridden]
  - **Purpose**: Assigns *bus* to the bus if the address lines do not exceed the memory locations.
  - **Parameter(s)**:
    - *bus*: *BusCom* reference object.

- `configured() const` [overridden]
  - **Purpose**: Checks if the memory storage and the bus are configured.
  - **Return**: A true if memory storage is at least one and the bus is linked; otherwise, a false.
- `at(size_t idx)`
  `at(size_t idx) const` [protected]
  - **Purpose**: Retrieve *Word* element at index *idx*
  - **Parameter(s)**:
    - *idx*: An index.
  - **Exception**:
    - *Out-Of-Bound Error*: Occurs if index out of bound.
  - **Return**: A *Word* reference.
- `toString() const` [overridden]
  - **Purpose**: Constructs a string representation of the memory as a numbered list.
  - **Return**: A string