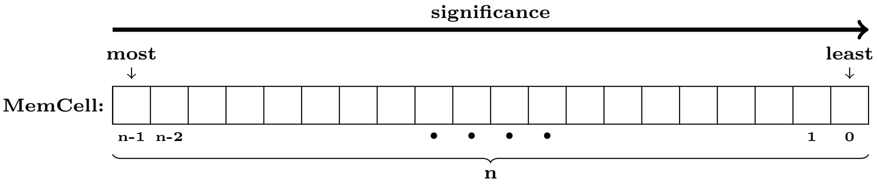
MemCell Class Documentation

Source File: ‘MemCell.h’
Namespace: ca
Class Header: `class MemCell : public Object`

Overview

The *MemCell* abstract class simulates a comparable binary word [most significant to least significant bit is read from left to right] that can be linked.



Constructors

- `MemCell()` (default constructor)
    - **Purpose**: Sets the cell’s identifier and makes it extendable and attachable.
- `MemCell(const MemCell& obj)` (copy constructor) [deleted]
- `MemCell(bool ext,bool att)` [protected]
    - **Purpose**: Sets the cell’s identifier and specifies its capacity for extension or attachment.
    - **Parameter(s)**:
        - *ext*: extendable flag
        - *att*: attachable flag

Destructor

- `~MemCell()` [virtual]
    - **Purpose**: Does nothing.

Assignment Operator

- `operator=(const Block& rhs)` [deleted]

Member Functions

- `size() const`
    - **Purpose**: Gets the length of the cell.
    - **Return**: The number of bits in the cell.
- `get(size_t idx) const`
    - **Purpose**: Gets a bit from the cell at a specified index
    - **Parameter(s)**:
        - *idx*: An index.
    - **Return**: The bit (Boolean) at the specified index.
- `set(size_t idx, bool bit)`
    - **Purpose**: Sets the cell’s bit at a specified index.
    - **Parameter(s)**:
        - *idx*: An index.
        - *bit*: A bit.
- `set(bool value = false)`
    - **Purpose**: Sets all cells to *value*
    - **Parameter(s)**:
        - *value*: A bit flag.

- `join(MemCell& obj)`
    - **Purpose**: Links *obj* to the cell if the cell has no attachment and *obj* is not attached.
    - **Parameter(s)**:
        - *obj*: A *MemCell* reference.

  `transfer(MemCell& obj)`
    - **Purpose**: Transfers the cell's content to *obj* up to the minimum size of both objects if *obj* is different from the cell.
    - **Parameter(s)**:
        - *obj*: A *MemCell* reference.
- `release()`
    - **Purpose**: Removes the cells attachment.
- `extended() const`
    - **Purpose**: Checks if the cell is extended.
    - **Return**: True if the cell has an attachment; otherwise, false.
- `appended() const`
    - **Purpose**: Checks if the cell is an attachment.
    - **Return**: True if the cell is an attachment; otherwise, false.
- `extendable(bool ext)` [protected]
    - **Purpose**: Sets the cell's ability to be extended
    - **Parameter(s)**:
        - *ext*: A extendable flag.
- `attachable(bool att)` [protected]
    - **Purpose**: Sets the cell's ability to be an attachment
    - **Parameter(s)**:
        - *att*: A attachable flag.
- `valid() const` [protected]
    - **Purpose**: Checks if the cell has an attachment.
    - **Return**: True if it has an attachment; otherwise, false.
- `node()`
  `node() const` [protected]
    - **Purpose**: Gets a reference to the attachment if the cell has an attachment; otherwise, a reference to the cell
    - **Return**: A *MemCell* reference
- `length() const` [private pure virtual]
    - **Purpose**: Gets the size of the cell without its attachment.
    - **Return**: An unsigned integer
- `read(size_t idx) const` [private pure virtual]
    - **Purpose**: Gets the bit of the cell at the specified index *idx*, excluding any attachment.
    - **Parameter(s)**:
        - *idx*: An index.
    - **Return**: A bit (Boolean).
- `write(size_t idx,bool bit)` [private pure virtual]
    - **Purpose**: Sets the bit of the cell at the specified index *idx*, excluding any attachment, to *bit*.
    - **Parameter(s)**:
        - *idx*: An index.
        - *bit*: A Boolean.

Non-Member Functions

- `operator==(const MemCell& lhs,const MemCell& rhs)`
  - **Purpose**: Check if *lhs* and *rhs* have equal values.
  - **Parameter(s)**:
    - *lhs*: A constant *MemCell* reference object.
    - *rhs*: A constant *MemCell* reference object.
  - **Return**: A true only if the values are equal; otherwise, a false.
- `operator!=(const MemCell& lhs,const MemCell& rhs)`
  - **Purpose**: Check if *lhs* and *rhs* have different values.
  - **Parameter(s)**:
    - *lhs*: A constant *MemCell* reference object.
    - *rhs*: A constant *MemCell* reference object.
  - **Return**: A true only if the values are different; otherwise, a false.
- `value(const MemCell& itm)`
  - **Purpose**: Provides conversion of a *MemCell* object to unsigned integer.
  - **Parameter(s)**:
    - *itm*: A constant *MemCell* reference object.
  - **Return**: The unsigned integer conversion of *itm* if the value of *itm* does not exceed $2^{32} - 1$; otherwise, zero.