```
architecture ram_arch of ram is
    type words_array is array (0 to n_words - 1) of std_logic_vector(words_size-1  downto 0);
    signal words: words_array;
begin
    process(read, address, reset) is
    begin
        if ((read'event and read='1') or (address'event and read='1')) then
            output_data <= words(address);
        end if;
    end process;
    process(write, address, reset) is
    begin
        if (reset = '1') then
            for i in 0 to n_words-1 loop
                words(i) <= std_logic_vector(to_unsigned(i, words(i)'length));
            end loop;
        elsif ((write'event and write='1') or (address'event and write='1')) then
            words(address) <= input_data;
        end if;
    end process;
end ram_arch ;
```

```
process is
begin
    r_input_data <= X"AA";
    r_write <= '1';
    r_address <= 0+3;
    wait for 25ns;

    r_input_data <= X"00";
    r_write <= '0';
    r_read <= '1';
    wait for 25ns;

    r_address <= 10;
    r_input_data <= X"00";
    r_write <= '0';
    r_read <= '1';
    wait for 25ns;

    r_input_data <= X"00";
    r_write <= '1';
    r_read <= '0';
    wait for 25ns;

    r_input_data <= X"00";
    r_write <= '0';
    r_read <= '1';
    wait for 25ns;
```