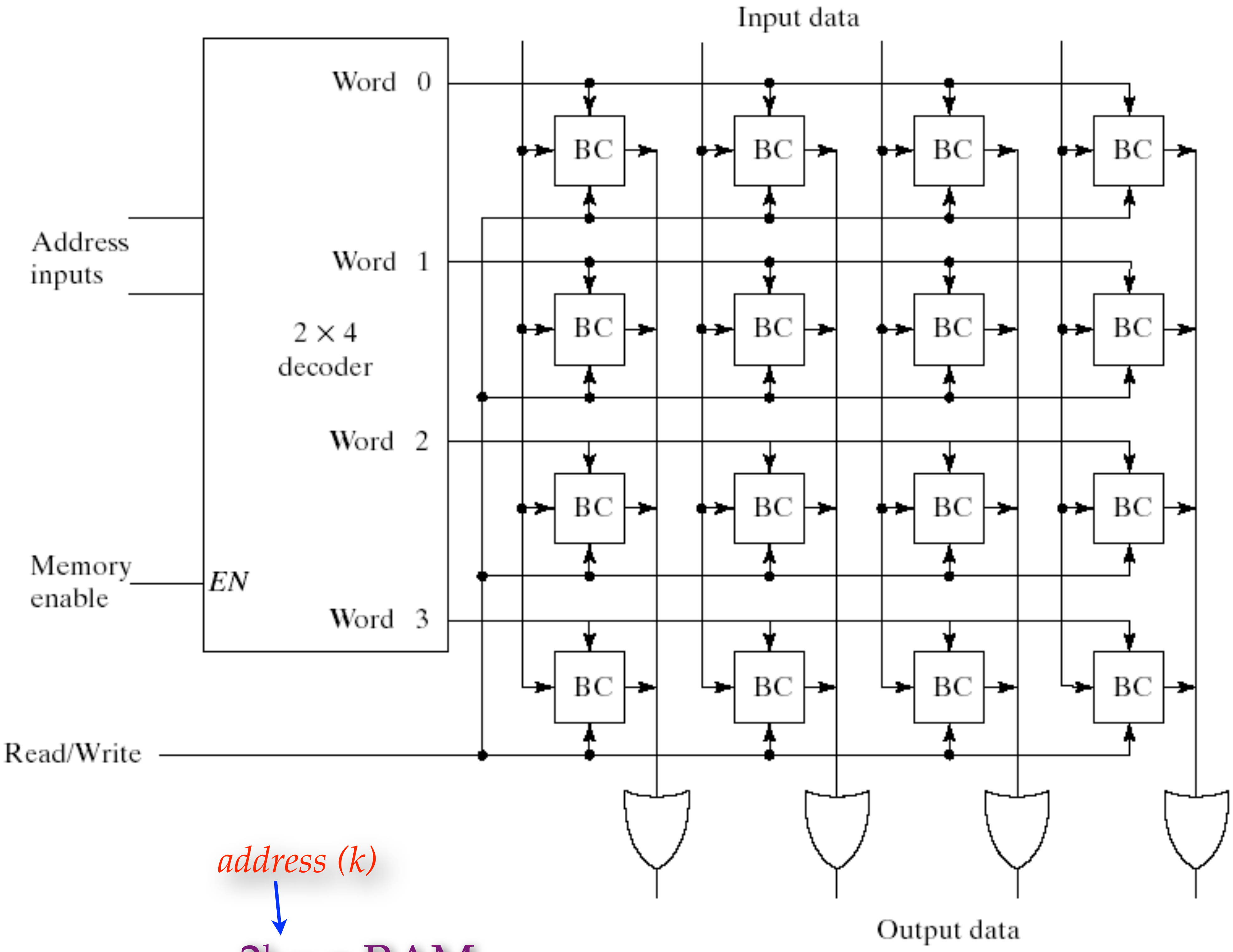
VGA Memory Mapping

Hsi-Pin Ma

http://lms.nthu.edu.tw/course/43639
Department of Electrical Engineering
National Tsing Hua University

RAM Revisit

2^2 x 4 RAM



address (k)

2^k x n RAM

# of output bits (n)

COE Format

- COE: memory coefficient file
- Two parameter:
  - **memory_initialization_radix**
    - Radix of the values in the *memory_initialization_vector*
    - Ex: 2, 10, or 16
  - **memory_initialization_vector:**
    - Memory content
    - Memory words are separated by whitespace
    - You can use comma (,) to help identify the boundary
    - Vector (entire memory) ended by semicolon

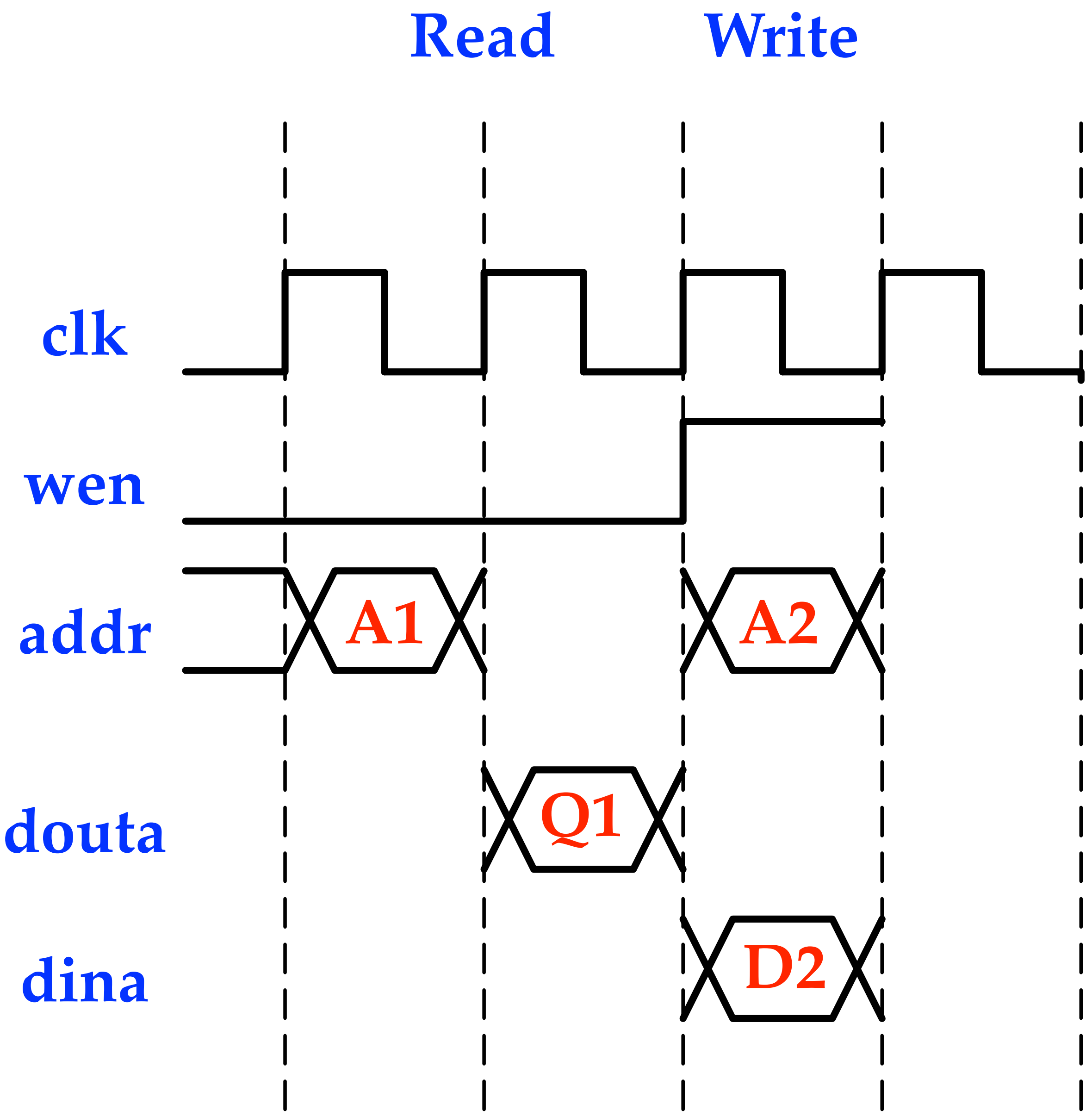
How to Use RAM Module

- Memory Read and Write is controlled by
  - wen pin: wen=0 for “read” and wen=1 for “write”
  - Separate read-address generator and write-address generator
- To refresh the image, you can
  - Always read out image from memory (wen=0) and display image in VGA synchronized with hsync and vsync. Write in image changes (wen=1) when necessary

How to Use RAM Module

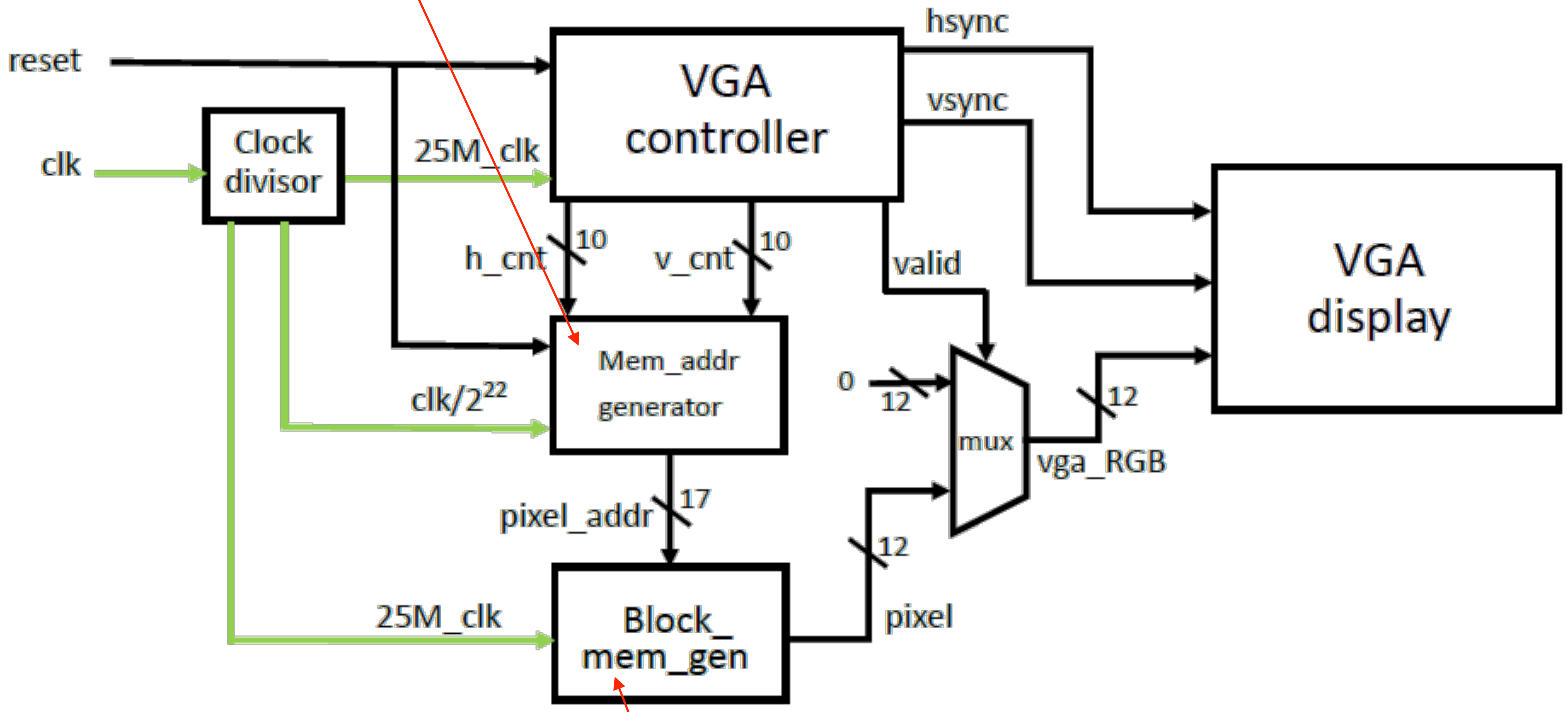
```
wire clk;
wire wen;
wire [63:0] data_in;
wire [63:0] out_64;
wire [5:0] addr;

RAM R1(
  .clka(clk),
  .wea(wen),
  .addra(addr),
  .dina(data_in),
  .douta(out_64)
);
```



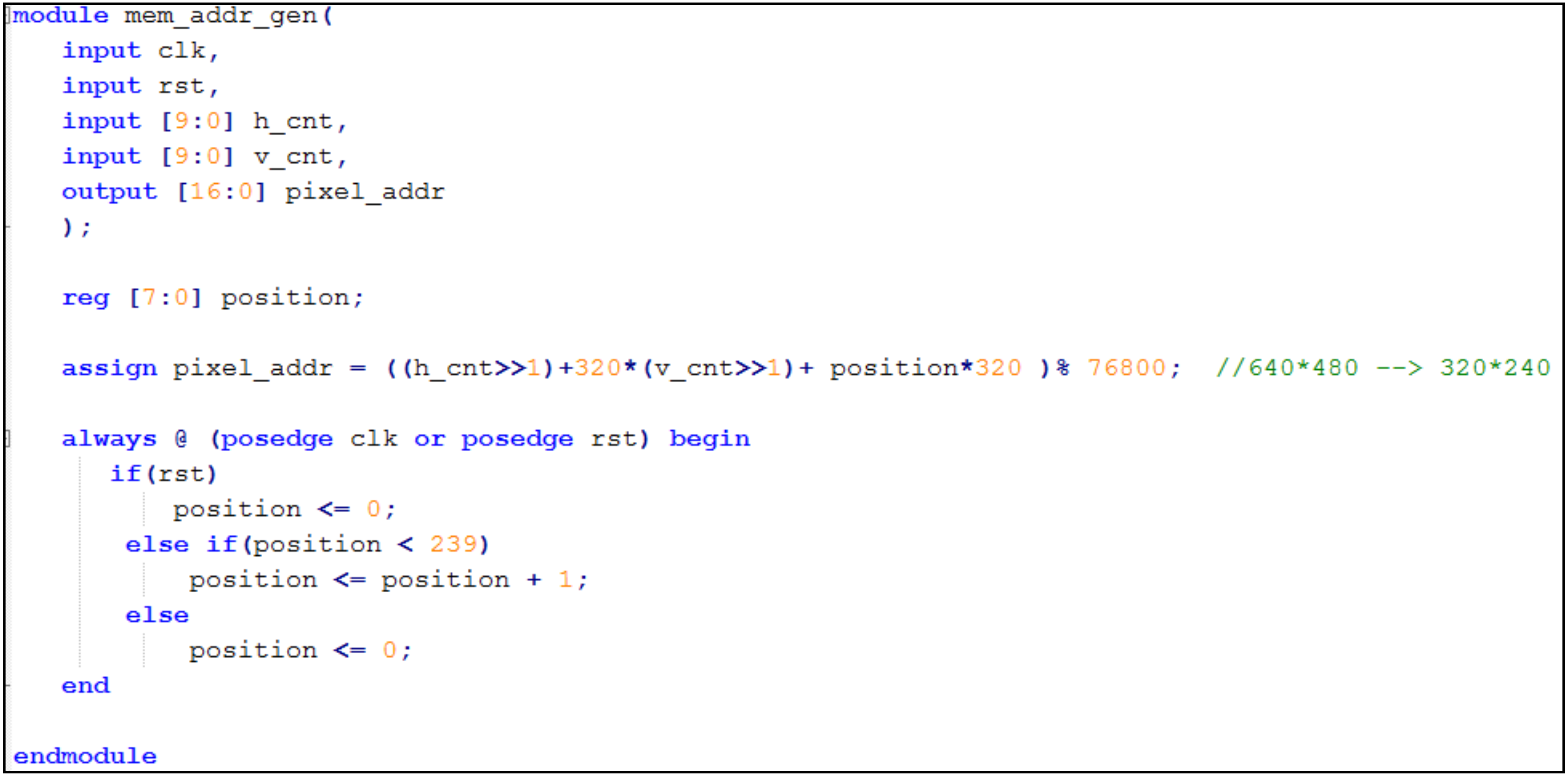
Demo 2: Block Diagram

reduce the resolution from 640x480 to 320x240



Saved Picture Access

Memory address generator



```
module mem_addr_gen(
   input clk,
   input rst,
   input [9:0] h_cnt,
   input [9:0] v_cnt,
   output [16:0] pixel_addr
   );

   reg [7:0] position;

   assign pixel_addr = ((h_cnt>>1)+320*(v_cnt>>1)+ position*320 )% 76800;  //640*480 --> 320*240

   always @ (posedge clk or posedge rst) begin
      if(rst)
          position <= 0;
       else if(position < 239)
           position <= position + 1;
       else
           position <= 0;
   end

endmodule
```