

PACMAN GAME

CSEE4840 Embedded System Design

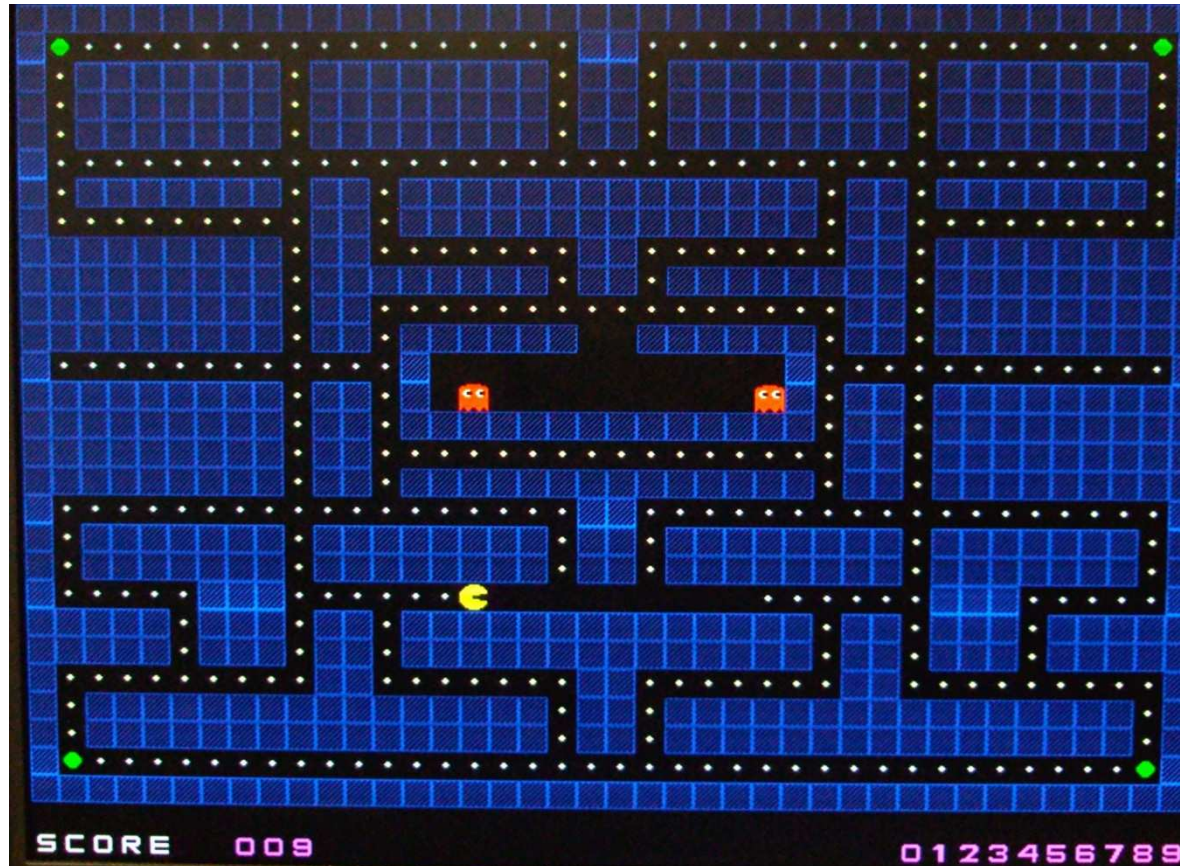
Advised by: Prof. Stephen Edwards

Dr. Baolin Shao, Dr. Scott Schuff




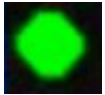

Presented by: Yunde Shi

Date: 1:30PM May 14th, 2010

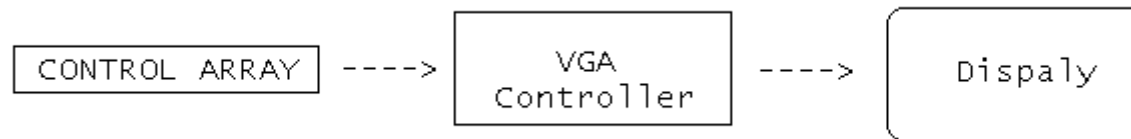
General Appearance



Basic Objectives to Achieve

- (1) There is only one PacMan; 
- (2) There is only one “active” ghost ; 
- (3) There are lots of small dots to eat (1 point/small dot); 
- (4) There are 4 large dots to eat (5 points/ large dot); 
- (5) The “score” at the bottom row should update whenever new dots are eaten; 
- (6) The “chasing” algorithm is the most naive ---- ghost simply goes towards the location of pacman.

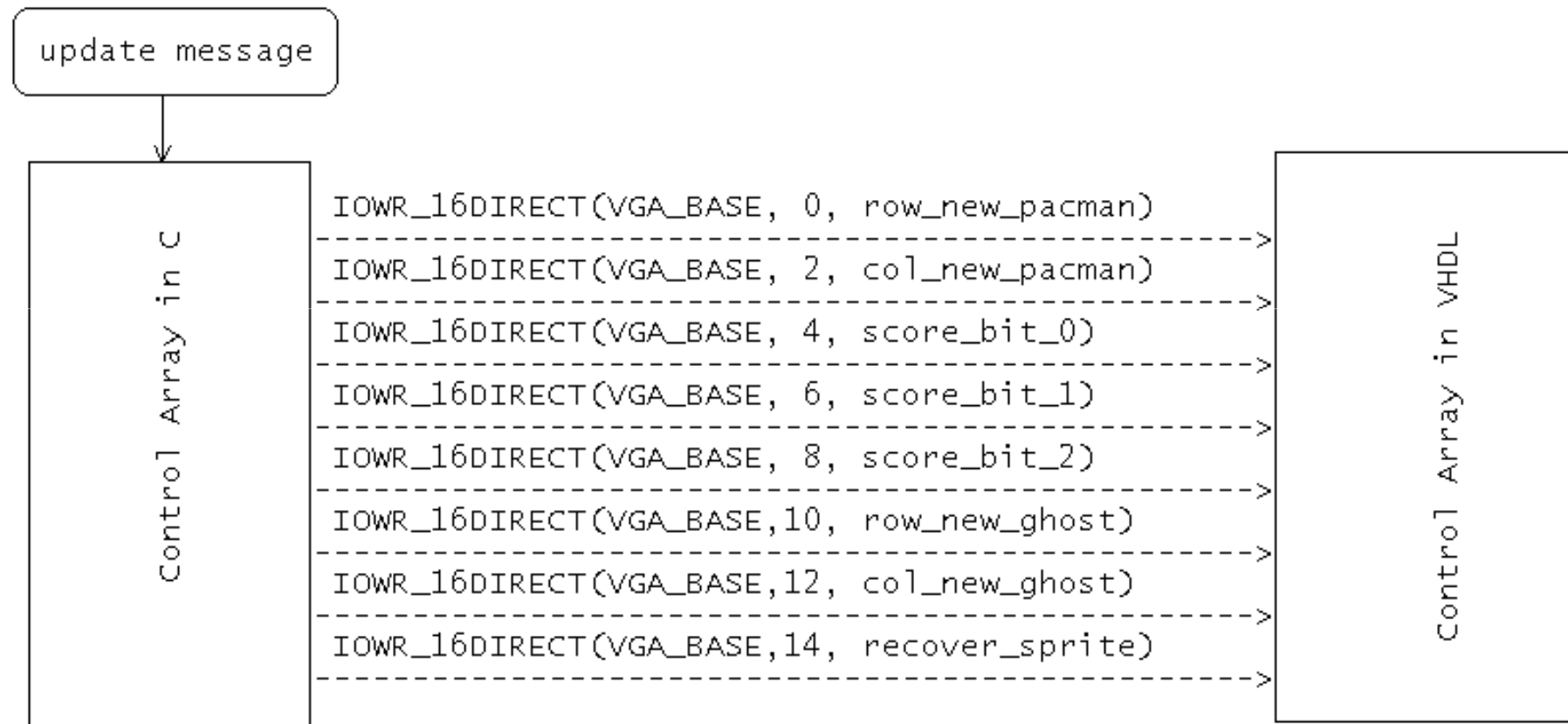
Basic Idea of the Game



*** TRICKS ***

No SRAM required for control array: the on-block ram is ~ 60KB which is “more than enough” ----- the control array in vhdl contains only totally $30 \times (40 \times 4) = 4800$ zeros and ones with a few additional sprites array 😊.

How to update the control array in both C and VHDL



DIFFICULTIES

(1) *PS2 reading never worked:*

Solution: To be able to distinguish the name of DAT from DATA.

(2) *The key press does not update the pacman location “sometimes”:*

Solution: keep sending the control array updates instruction even in the while loop.

(3) *How to have more sprites to display, e.g. “score”:*

Solution: using additional constraints of row number.

```
-- for control_array_V = 29:
-- 'S'           : 0001
-- 'C'           : 0010
-- 'O'           : 0011
-- 'R'           : 0100
-- 'E'           : 0101
-- '0'           : 0110
-- '1'           : 0111
-- '2'           : 1000
-- '3'           : 1001
-- '4'           : 1010
-- '5'           : 1011
-- '6'           : 1100
-- '7'           : 1101
-- '8'           : 1110
-- '9'           : 1111

-- for control_array_V <= 27:
-- pacman       : 0001
-- brick        : 1111
-- background   : 0000
-- dot_small    : 0010
-- dot_large    : 0011
-- ghost        : 0100
```

