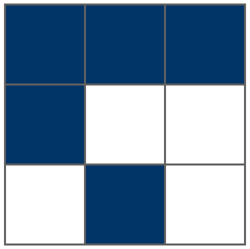
The Game of Life



Steven Chen, Juan Gutierrez, Vincenzo Zarrillo
{stc2104, jmg2048, vaz2001}@columbia.edu

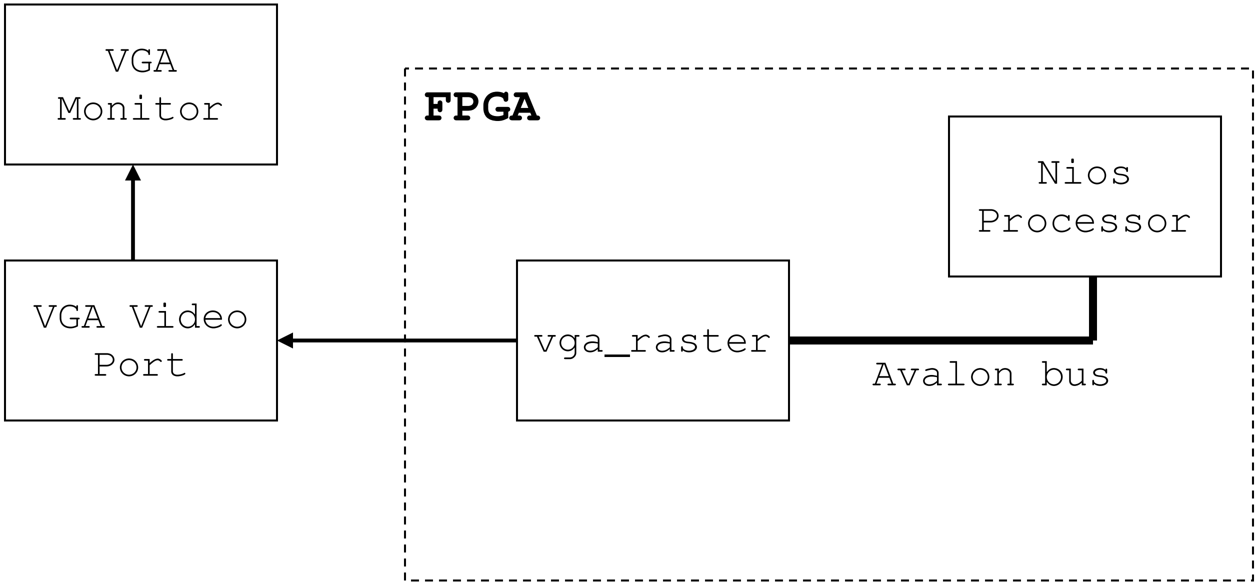
Embedded Systems Design, CS 4840

May 8, 2007

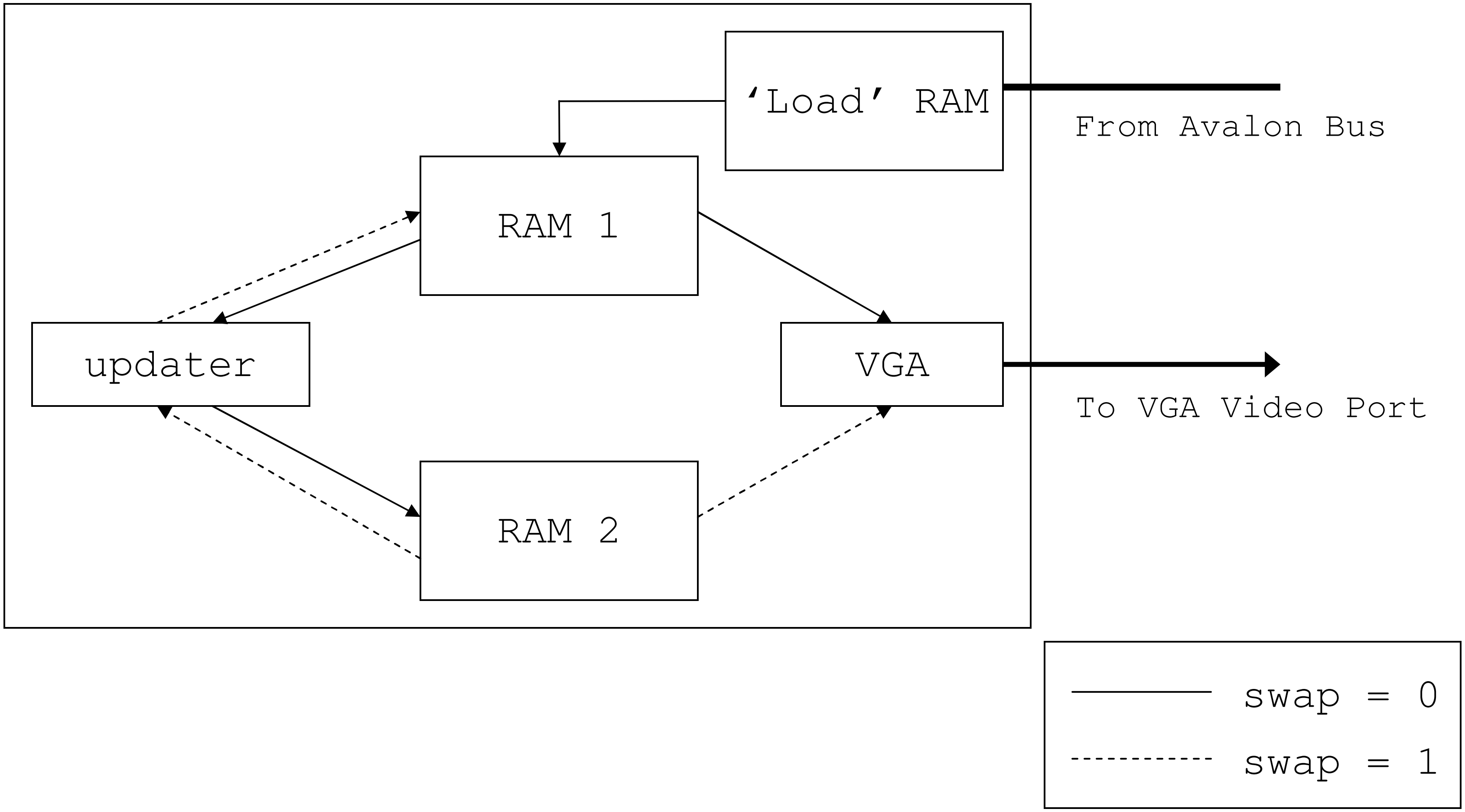
Overview of the System

- Based on initial coordinates, outputs to VGA the game visualization (the ‘board’)
    - White indicates ‘life’ and blue indicates ‘death’
- Each organism modeled as one pixel on a 256x256 pixel board
- Generations occur roughly every second
- Hardware used to update each generation of the game
- Software (C program) used to pass to hardware the initial conditions of the board

Architectural Design



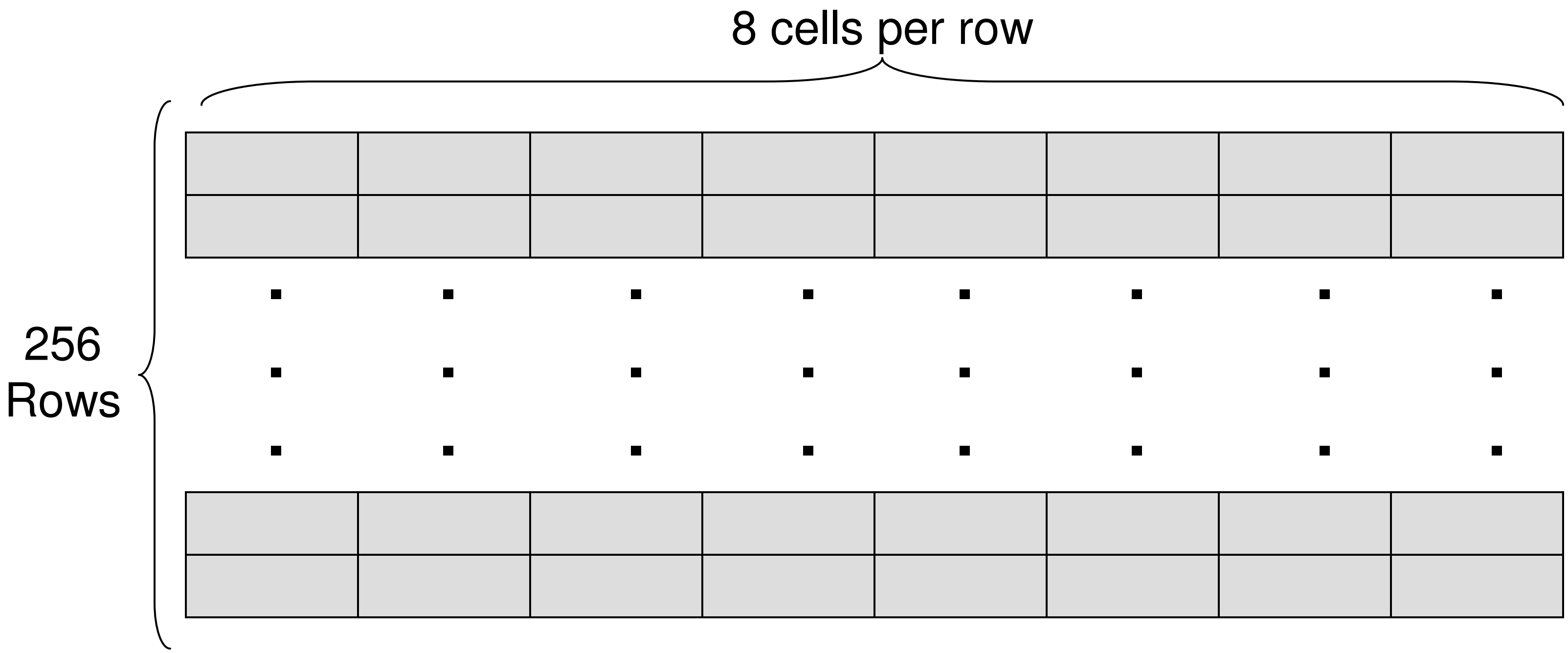
vga_raster Component Design



Overall System

- Nios sends initial coordinates to the ‘Load’ RAM through the Avalon bus
- ‘Load’ RAM contents loaded into RAM 1 (‘current’)
- VGA reads from ‘current’ while updater also reads from ‘current’ and writes to RAM 2 (‘next’)
- ‘next’ and ‘current’ are then swapped

Internal Representation of Game Board



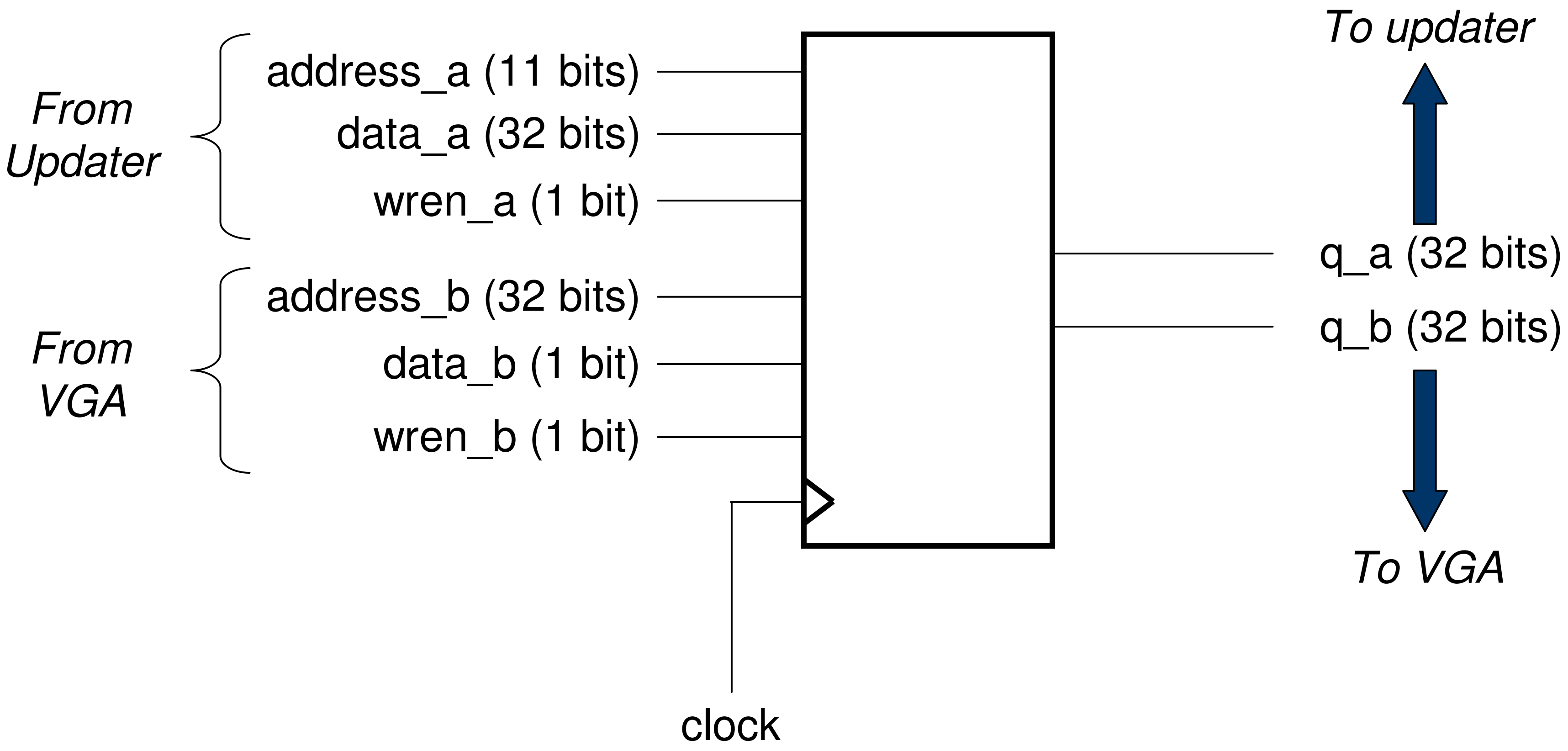
▨

Each ‘cell’ holds 32 bits

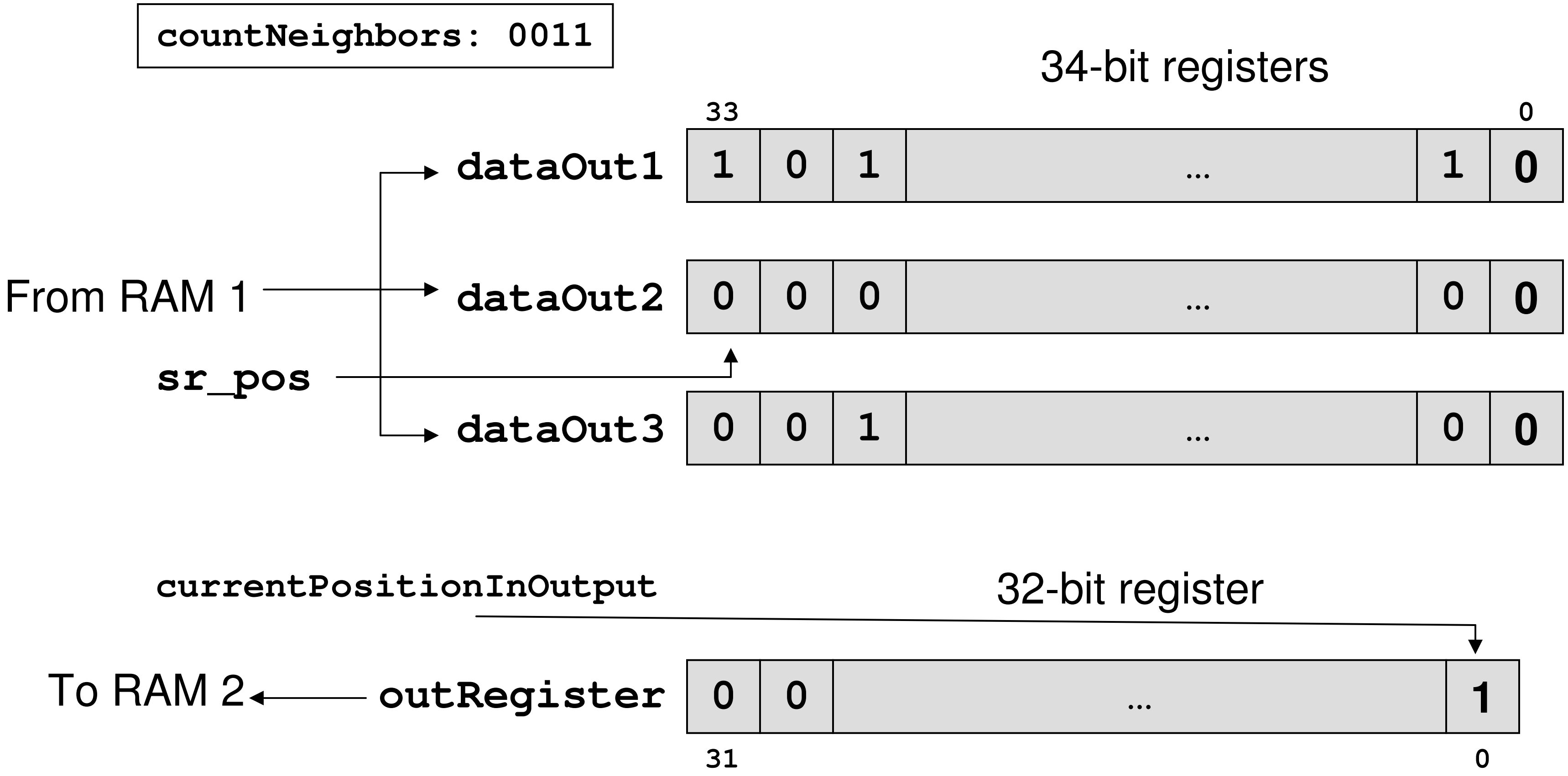
8 cells X 32 bits = 256 bits total per row

8 cells X 256 rows = 2048 (2^{11}) cells total in board

Dual-Port RAM - Current



Game Logic Implementation



countNeighbors: 0011

34-bit registers

	33					0
dataOut1	1	0	1	...	1	0
dataOut2	0	0	0	...	0	0
dataOut3	0	0	1	...	0	0

From RAM 1

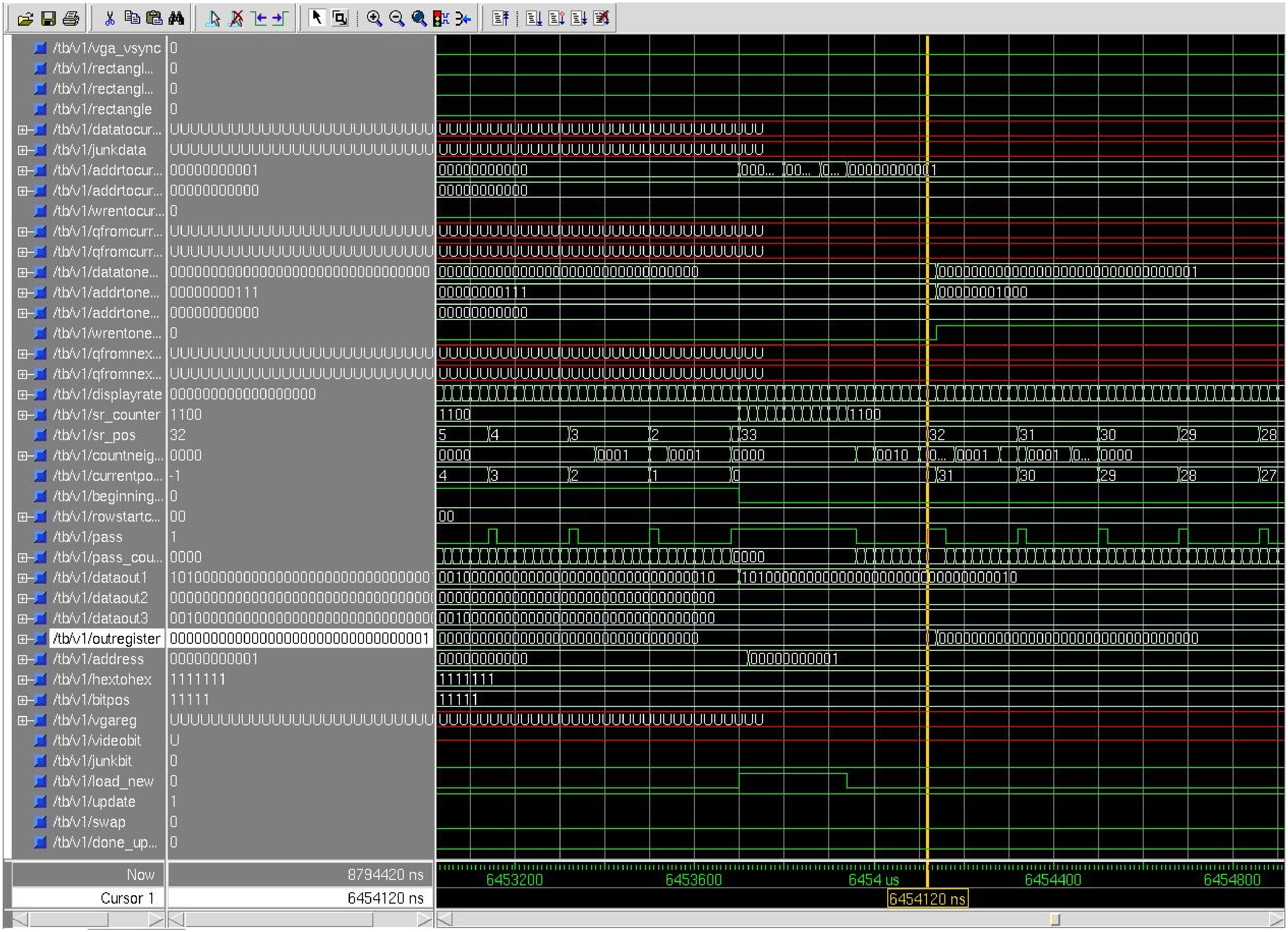
sr_pos

currentPositionInOutput

32-bit register

	31			0
outRegister	0	0	...	1

To RAM 2



/tb/v1/vga_vsync	0
/tb/v1/rectangl...	0
/tb/v1/rectangl...	0
/tb/v1/rectangle	0
/tb/v1/datatocur...	UUUUUUUUUUUUUUUUUUUUUUUUUUUUU
/tb/v1/junkdata	UUUUUUUUUUUUUUUUUUUUUUUUUUUUU
/tb/v1/addrtocur...	00000000001
/tb/v1/addrtocur...	00000000000
/tb/v1/wrentocur...	0
/tb/v1/qfromcurr...	UUUUUUUUUUUUUUUUUUUUUUUUUUUUU
/tb/v1/qfromcurr...	UUUUUUUUUUUUUUUUUUUUUUUUUUUUU
/tb/v1/datatone...	00000000000000000000000000000000
/tb/v1/addrtone...	00000000111
/tb/v1/addrtone...	00000000000
/tb/v1/wrentone...	0
/tb/v1/qfromnex...	UUUUUUUUUUUUUUUUUUUUUUUUUUUUU
/tb/v1/qfromnex...	UUUUUUUUUUUUUUUUUUUUUUUUUUUUU
/tb/v1/displayrate	000000000000000000
/tb/v1/sr_counter	1100
/tb/v1/sr_pos	32
/tb/v1/countneig...	0000
/tb/v1/currentpo...	-1
/tb/v1/beginning...	0
/tb/v1/rowstartc...	00
/tb/v1/pass	1
/tb/v1/pass_cou...	0000
/tb/v1/dataout1	1010000000000000000000000000000000
/tb/v1/dataout2	0000000000000000000000000000000000
/tb/v1/dataout3	0010000000000000000000000000000000
/tb/v1/outregister	00000000000000000000000000000001
/tb/v1/address	00000000001
/tb/v1/hextohex	1111111
/tb/v1/bitpos	11111
/tb/v1/vgareg	UUUUUUUUUUUUUUUUUUUUUUUUUUUUU
/tb/v1/videobit	U
/tb/v1/junkbit	0
/tb/v1/load_new	0
/tb/v1/update	1
/tb/v1/swap	0
/tb/v1/done_up...	0
Now	8794420 ns
Cursor 1	6454120 ns

VGA Implementation

- Reads bit by bit and colors pixel accordingly
- After reading bottom right end of the board, updater turns on

Nios Implementation

- Writes 32 bits to each location in RAM
- Random set of numbers or hard-coded set of numbers as initial conditions

Implementation Experiences and Issues

- Necessity of Precise Timing
- Difficulties in deciding on best and easiest implementation of game logic
  - Shift registers, components, etc.
- Writing Initial Conditions into the program
  - Issues with addressing
- Reading from a file in C

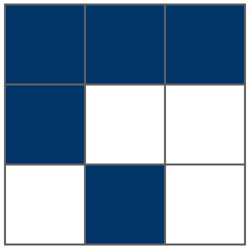
Primary Roles

- Steve
  - Updater Implementation
  - Design Document, Final Report, Presentation
- Juan
  - Updater/VGA/Nios Implementation
  - System Integration
- Vinny
  - VGA/Nios Implementation
  - System Integration
- Everyone
  - Design, Debugging, Troubleshooting

Lessons Learned

- Timing Diagrams – Draw them first!
- Test every potential thing that could go wrong as soon as you can.
- The simulator is your best friend
- Think Hardware, not Software
- It's never too early to start

The Game of Life



Steven Chen, Juan Gutierrez, Vincenzo Zarrillo
{stc2104, jmg2048, vaz2001}@columbia.edu

Embedded Systems Design, CS 4840

May 8, 2007