PONG

Group 27:

Michael Acquaviva, Andrew Moser

Antoine Vilain, Alan Cao

PROJECT TEAM

ANTOINE

MICHAEL

ALAN

ANDREW

PROJECT DESCRIPTION

- PONG game: 2 players each control a paddle and bounce a ball, attempting to score on the opposing player's goal, winning at 7 points
- Additional features added:
    - Paddle size powerup
    - Player vs Computer (varying difficulties)

PROJECT INITIAL GOALS

USER INPUTS

- Paddle moves in accordance with user's joystick input
- Game menu navigated with user's keyboard input
- ...

GAME LOGIC

- Ball bounces off paddle at right angle
- Ball moves in straight line until collision
- Point scored when ball hits end wall
- Game ends when score of 7 reached
- ...

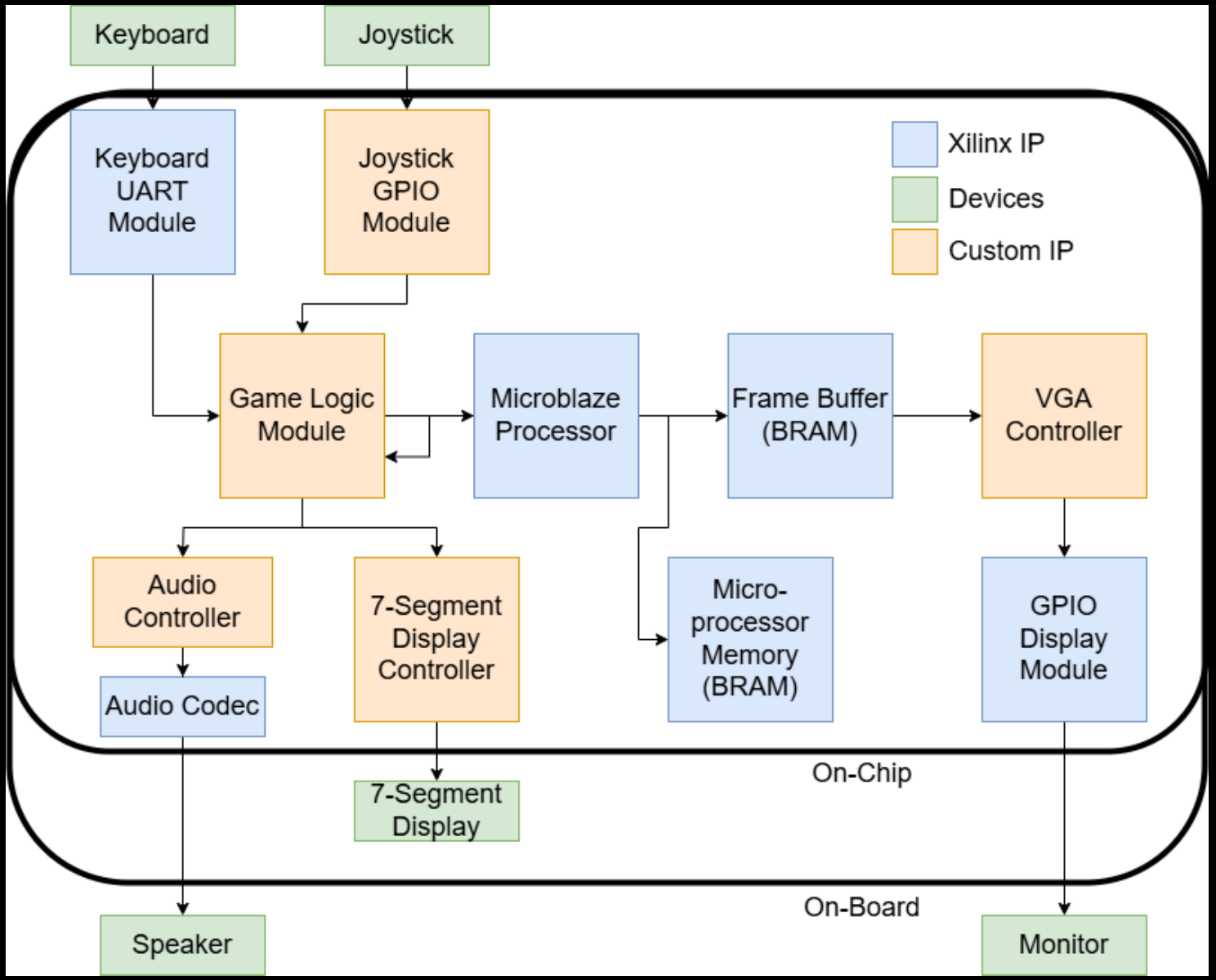
OUTPUT PERIPHERALS

- Screen displays output in smooth frames
- Speaker outputs game's sound effects
- 7-segment display accurately shows score
- ...

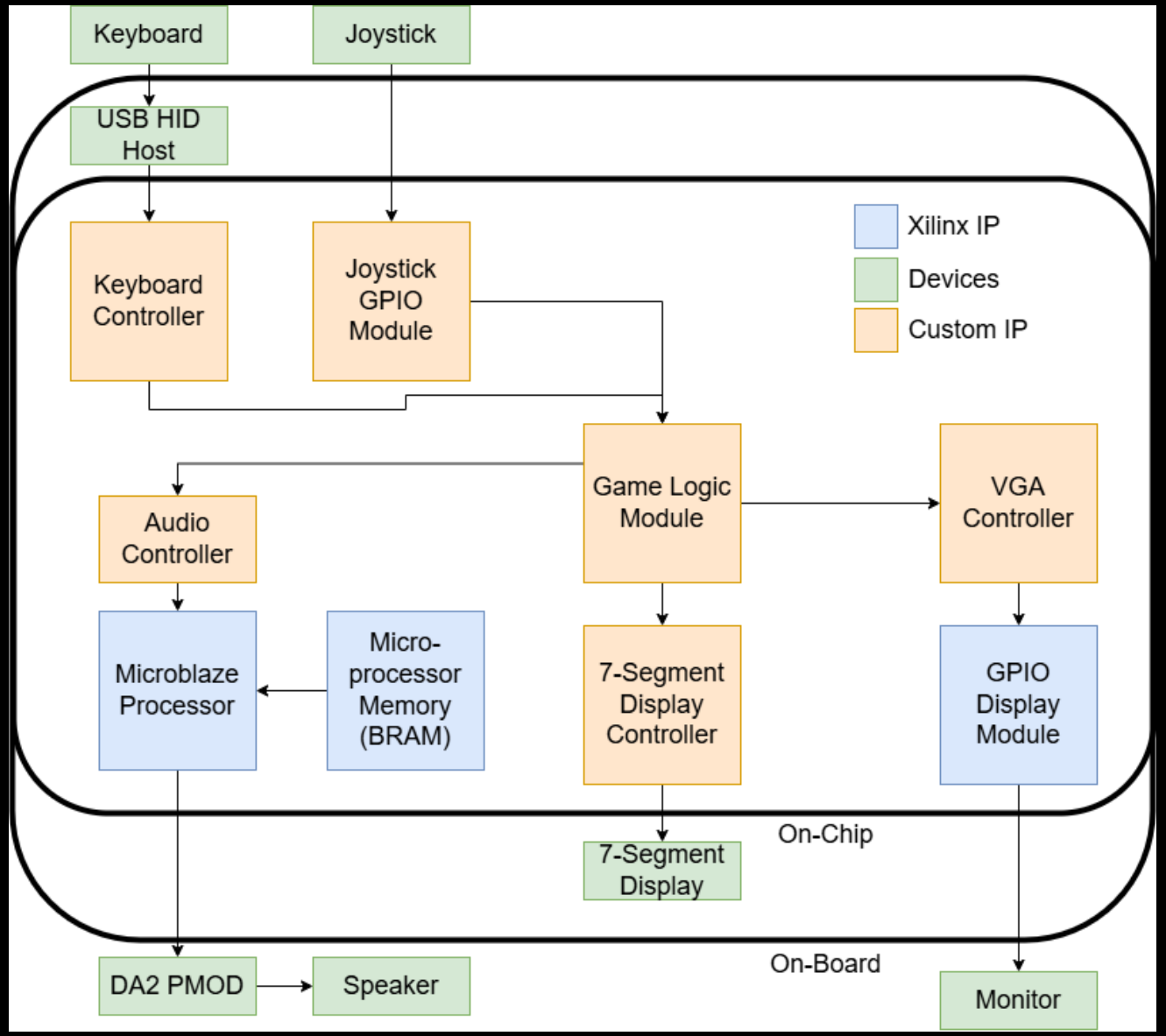
IMPLEMENTATION OVERVIEW

- Joysticks to grab user inputs
- Game logic all done in hardware except for microblaze which drives the sound output through the speaker
- VGA display outputs the game
- 7 seg display for score
- Speaker output for sound effects
- Keyboard for resets

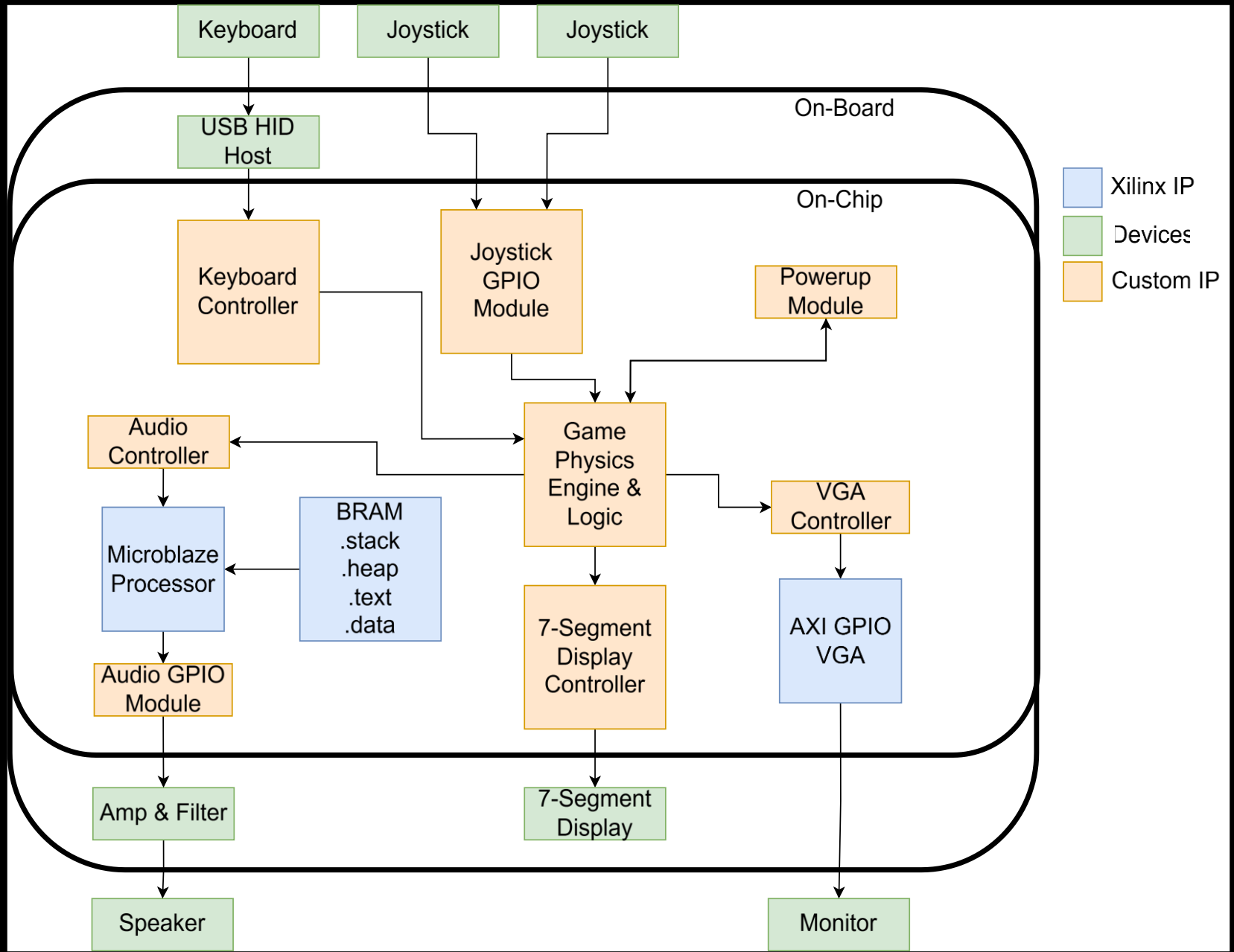
IMPLEMENTATION – 1ST BLOCK DIAGRAM



IMPLEMENTATION – 2ND BLOCK DIAGRAM



FINAL IMPLEMENTATION



HARDWARE OVERVIEW: SYSTEM COMPONENTS

- Nexys4 DDR FPGA Board
- PMOD JSTK2 x2
- USB Keyboard
- Speaker
- Display Monitor

COMPONENTS – INPUTS

- PMOD JSTK2

  SPI Interface: FSM to control bit-level transmission using MOSI, MISO, SCLK in SPI mode 0

  SPI Controller: FSM to control high-level transaction, managing the 5-byte data packet shift register

  Joystick: Extracts X positions and output up/down control signals to game logic

  - Threshold based on y position generates 2b up/down/neutral signal, x position ignored

- USB Keyboard (FPGA board microcontroller emulates PS/2 bus)

  PS2 Controller: Reads incoming scancodes and stores them in a 32-bit shift register

  Keyboard: Case statement triggers output flags for appropriate scancodes

  - Detects spacebar being released for game start and 'r' being released for reset

COMPONENTS – GAME LOGIC

- PONG

  PONG: Computes game state at 30 FPS, including paddle and ball positions, ball and wall/paddle collisions, ball spin, scoring and ball reset, game-over

  Paddle Size Powerup: Spawns powerup in pseudo-random time and position, sets paddle heights

COMPONENTS – OUTPUTS

- Display Monitor

  VGA Sync: Generates VGA synchronization signals for 640 x 480 resolution

  VGA Display: Generates VGA output signals based on FSM of PONG game states

- Seven Segment Display

  Binary to Decimal: Converts player score into binary-coded decimal notation

  Scoreboard: Outputs binary-coded decimal scores on seven-segment display

COMPONENTS – AUDIO AND SOFTWARE OUTPUT

- Audio Output utilized Microblaze
- Amplifier and low pass filter
- Used Square Wave audio
- Majority of RAM space
- Kept duty cycle 50% since choose to not distort sound

PROJECT COMPLEXITY

Item	Complexity
Transfer data from desktop to FPGA using UART + MicroBlaze + stdin	0
Hardware	
VGA Output without MicroBlaze Involvement	1.0
USB Keyboard Implementation	0.5
7 Segment Display	0.2
2-Axis Joystick PMOD	0.75
Algorithm Complexity (PONG + paddle powerup + AI)	1.0
On-board Audio Output Port	0.5
Software	
Algorithm Complexity	0.5
Total	4.45

RESOURCE USAGE

- 10% LUT usage
- 74% Block RAM usage
- 1% DSP usage

1. Slice Logic

Site Type	Used	Fixed	Available	Util%
Slice LUTs	6348	0	63400	10.01
LUT as Logic	5979	0	63400	9.43
LUT as Memory	369	0	19000	1.94
LUT as Distributed RAM	96	0		
LUT as Shift Register	273	0		
Slice Registers	4801	0	126800	3.79
Register as Flip Flop	4797	0	126800	3.78
Register as Latch	0	0	126800	0.00
Register as AND/OR	4	0	126800	<0.01
F7 Muxes	211	0	31700	0.67
F8 Muxes	0	0	15850	0.00

3. Memory

Site Type	Used	Fixed	Available	Util%
Block RAM Tile	100	0	135	74.07
RAMB36/FIFO*	100	0	135	74.07
RAMB36E1 only	100			
RAMB18	0	0	270	0.00

4. DSP

Site Type	Used	Fixed	Available	Util%
DSPs	2	0	240	0.83
DSP48E1 only	2			

FUTURE DIRECTIONS

- Add colour to VGA Output
- Paddles can move horizontally (but restricted to their side of field, like air hockey)
- More powerups (invisible ball, inverted controls, multiple balls, shield, etc.)
- Better way to produce audio output

FINAL PROJECT DEMO