Chess Teacher

Team B4: Michael Cai, Joseph Chang, Jee Woong Choi

Use case

- Learn how to play Chess against an AI
  - Not having to play with a human coach
    - Saves money from in-person tutoring
    - Social distancing
  - Customized levels depending on progress
- Play over the board with physical pieces rather than on computer
  - Creates a more realistic environment
  - Simulates tournament or competitive setting
- Analyze your games, showing various moves in a given turn
- Areas Covered:
  - Software Systems, Signals and Systems, Hardware Systems

Requirements (Game)

- Gameplay mechanics:
  - Human must wait until his move is registered on UI and computer responds
  - UI shows computer move, human must move for computer
  - UI is able to recommend a move if user desires
- One player plays at a time
- Camera has to be set up on top of the board with a clear view
- Camera has to recognize the board setup before the game
- Only valid moves will be accepted
- A move cannot be changed once placed

Requirements (Computer Vision)

- Use camera and computer vision to track:
  - Board
  - Pieces
  - Moves
- Can detect changes in board state when new pieces are placed within 400ms
  - Includes piece differentiation
- Move detection accuracy of 99 percent
- Decrease processing time by downscaling the image
- Avoids complex circuitry and hardware which would be embedded into board

Requirement (Hardware/FPGA)

- Parallelize the generation of all possible moves
  - Process all possible moves within 15 clock cycles, on 50MHz clock => 300 ns.
- Provide information back to CPU using UART communication.
  - Moves encoded in algebraic notation
  - IO communication of 20000bits/per board state
  - Using baud rate of 460800, 23 sets of moves can be sent each second

Requirements (User Interface)

- Easy to visualize moves for users
- Successful indication of users' moves and the computer's moves
- Correct Representation of the Chessboard status
- Clear indication of timer and turns
- Simple design to make UI intuitive
    - Buttons can toggle between analysis and non-analysis

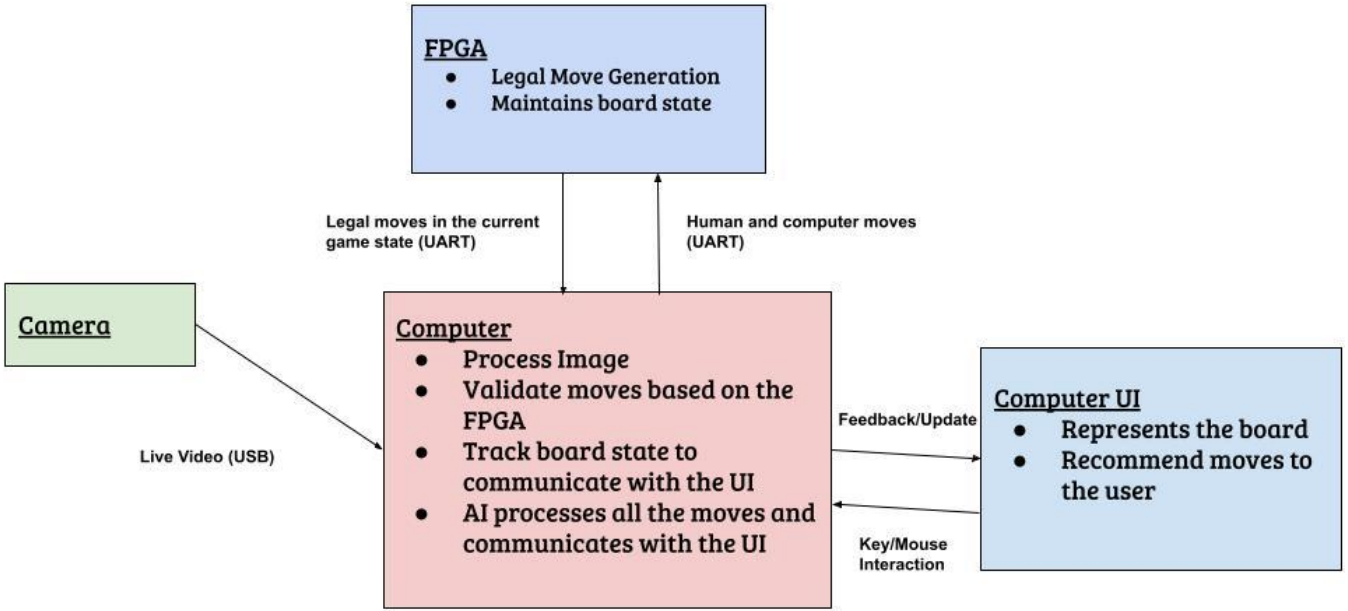
Technical challenges (Computer Vision)

- Processing time of image to detect moves
- Coordinating the components together, FPGA, screen, and camera
- Transferring the CV data to the UI
- Accuracy of the move detection
    - Position of the camera could affect the accuracy
    - Camera has to handle noise
    - Resample image if move is detected incorrectly

Technical challenges (FPGA)

- Parallelizing of the overall system
- Pipelining legal move generation to fit within clock period
- Efficient communication between FPGA and system
    - Keep response time low
    - IO is the bottleneck and efficient UART is needed
    - Must support high baud rate to offset IO bottleneck

Solution Approach



Testing, Verification and Metrics

Requirement	Testing Strategy	Metrics
Move detection	Software + Visual confirmation	99% accuracy in move detection & < 400 ms processing time
FPGA legal move generation	Hardware testbench (ensure correct legal moves generated)	100% Correct
Communication between Computer and FPGA	Hardware testbench (analyze packets are sent correctly)	Latency of < 1s & 100% data accuracy
UI	Visual confirmation	100% accuracy in representation of the board

Tasks and Division of Labor

CV:

- Detect pieces (Joseph & Jee Woong)
- Detect board (Joseph & Jee Woong)
- Ensure high level of correctness (Joseph)
- Detect moves (Joseph)

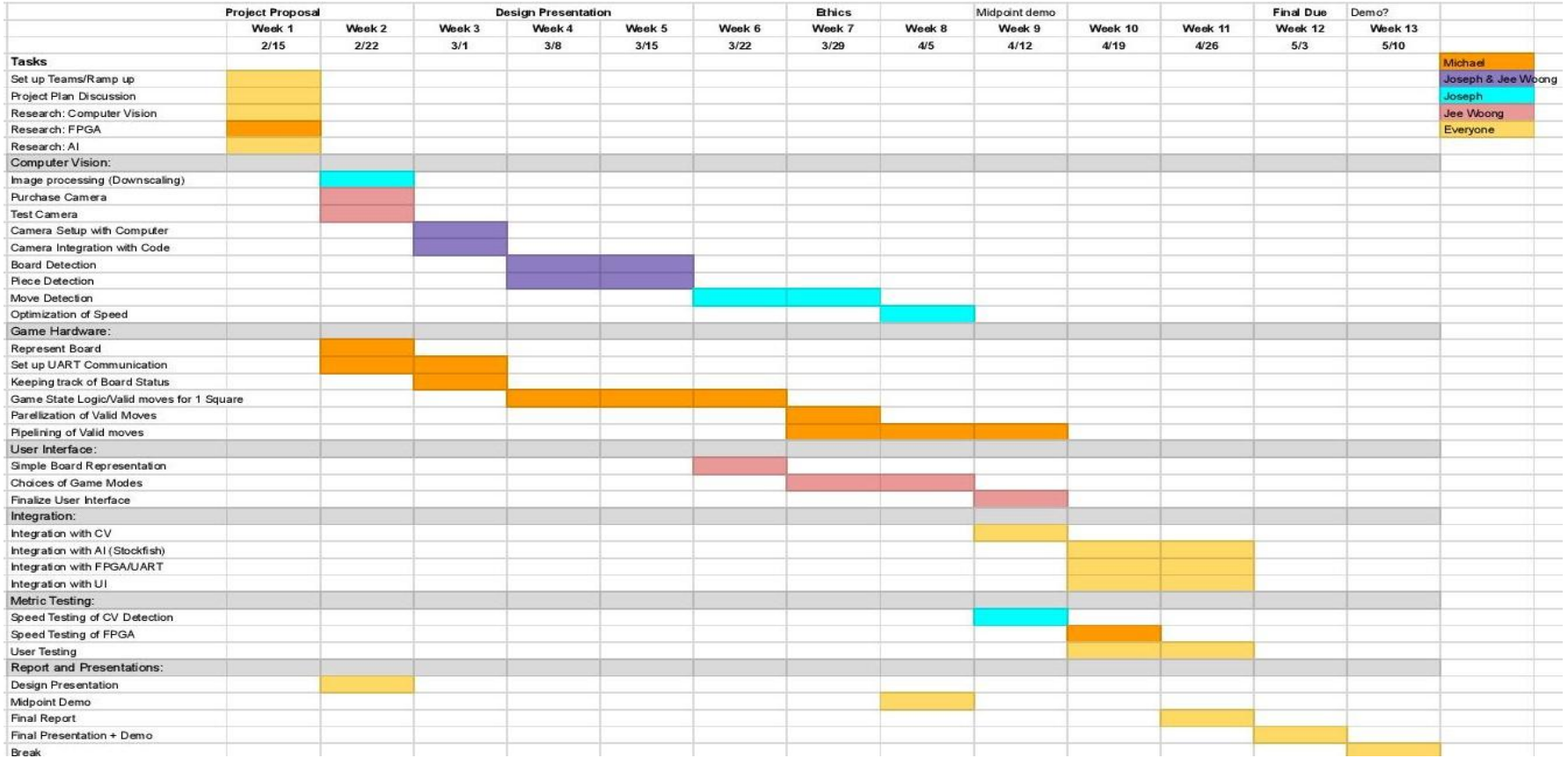
Game Logic/AI/UI:

- Gives player computer move (Jee Woong)
- Toggle between various move lines (Jee Woong)

FPGA:

- Legal move generation (Michael)
- Accelerates the game logic (Michael)
- Highly parallelizable on FPGA as each square can have its own legal move generation module (Michael)
    - Parallelize on all 64 game squares
- Learn how to communicate efficiently between the FPGA and CPU and vice-versa via UART (Michael)

Schedule



	Project Proposal		Design Presentation				Ethics		Midpoint demo			Final Due	Demo?	
	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	
	2/15	2/22	3/1	3/8	3/15	3/22	3/29	4/5	4/12	4/19	4/26	5/3	5/10	
Tasks														Michael
Set up Teams/Ramp up														Joseph & Jee Woong
Project Plan Discussion														Joseph
Research: Computer Vision														Jee Woong
Research: FPGA														Everyone
Research: AI														
Computer Vision:														
Image processing (Downscaling)														
Purchase Camera														
Test Camera														
Camera Setup with Computer														
Camera Integration with Code														
Board Detection														
Piece Detection														
Move Detection														
Optimization of Speed														
Game Hardware:														
Represent Board														
Set up UART Communication														
Keeping track of Board Status														
Game State Logic/Valid moves for 1 Square														
Parellization of Valid Moves														
Pipelining of Valid moves														
User Interface:														
Simple Board Representation														
Choices of Game Modes														
Finalize User Interface														
Integration:														
Integration with CV														
Integration with AI (Stockfish)														
Integration with FPGA/UART														
Integration with UI														
Metric Testing:														
Speed Testing of CV Detection														
Speed Testing of FPGA														
User Testing														
Report and Presentations:														
Design Presentation														
Midpoint Demo														
Final Report														
Final Presentation + Demo														
Break														