Project Title: Dance Invaders – A Four-Monitor Cabinet that echoes with the golden age of arcades	
Name: 張智孝	ID: 108062101
Name: 詹詠淇	ID: 108062204

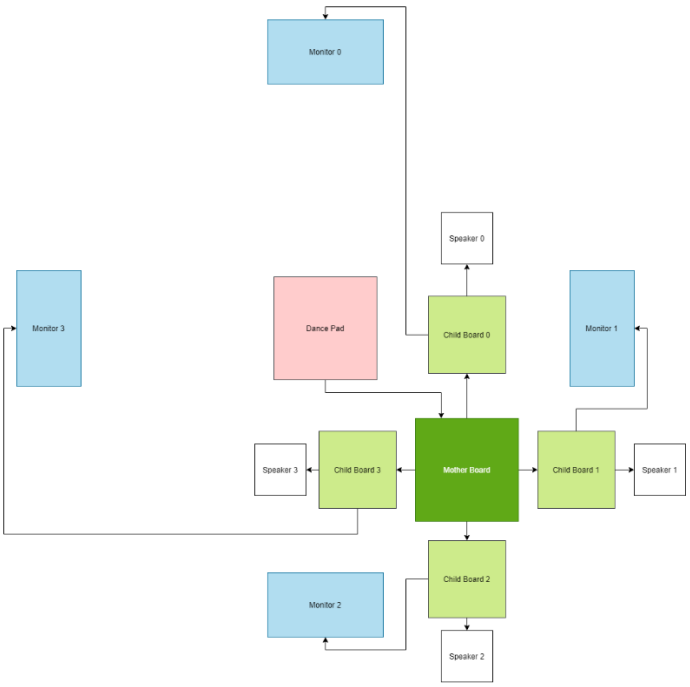
Project Description:

1. 概念圖、功能描述與使用到的 I/O Devices 或額外的機構設計

Devices needed:

- 5 Basys3 FPGA boards. 1 being the mother board, with the remaining being the child boards. (Consider 4 pairs of PmodDA1 modules?)
- 1 Dance pad. Mainly for the pressure-sensitive D-pad. The DIY options may be cheaper, but they can take up a huge amount of time in the development process. Can either go into the board's USB port, or decoded by an external computer. (Estimated Price: Lowest price spotted so far is NTD $169. Overall estimates are around $300)
- 4 monitors with VGA ports. During demo, the monitors found in EECS 236/238 will be used. Connected to the child boards.
- 4 speakers. Again, using the ones in EECS 236/238. (Or we can order PmodAMP3 modules)

Block Diagram:



User Instructions:
The player enters the booth, inserts his or her credit, and steps on any of the four arrows to begin. All aliens will appear at once, orbiting around the player from far away, though they are colored a dark gray and cannot be hit yet. At fixed time points, 1 to 8 aliens may move forward and enter firing range, in which they will alternate between orbiting and moving forward in fixed time intervals. Once the aliens get to the player, it's game over, so the player must step on the arrows to shoot down any alien that gets too close in time. When the player steps on an arrow, a laser is fired and will hit the first alien in its way.

The game speed increases whenever the remaining number of aliens reach certain thresholds. Once all aliens are shot down, the level increases, and the game starts over at a higher difficulty. When the player loses, their score is shown on the screen, and they can input their 3-character player tag using arrows to put the score on the scoreboard, which will persist until the game is soft-reset.

Demo Objectives: To successfully demonstrate the game experience we're aiming to achieve, and hopefully, make it complete enough for TAs and the professor to give it a try as well.

Items and Specifications:

1. Synchronization Protocol:
   The boards form a tree topology, with the mother board being the root and the child boards being the leaves. We will design the protocol interface between the root and its children, so that serial signals can be exchanged across the inherently different clock domains across boards.

1.1. Tentative Unit Test:
   Serialize a sequence of 8-bit unsigned integer signals from one board and send it to another. The other board accumulates the sequence of integers and then sends it back once the transmission is complete.

2. Dance Pad Interface:
   The Dance Pad is essentially a 3x3 matrix of pushbuttons with a unified output port. We will hook it up with the mother board's pins so that we can receive input from the Dance Pad.

2.1. Tentative Unit Test:
   Set the pins to be LED pins and see if the corresponding LED lights up when one of the four directional tiles is stepped on.

3. VGA Interface:
   We will use preexisting IP from the lab that uses VGA display for this item.

4. Audio Interface:

We will use preexisting IP from the lab that uses audio output for this item.

5. Hardcoded ROM:
   We will preprocess bitmaps and audio files into hex code and hardcode them. If necessary, run-length compression can be considered.
6. Game Logic for “Dance Invaders”, as an algorithmic state machine (below uses the term “scene” over “state” to fit in line with typical video game terminology):

6.1. Game Start Scene:
Every monitor should display the same start screen, along with the number of credits inserted (0 to 9). If the number of credits is greater than 0 and a directional pad is stepped on, one credit is consumed, and the game starts from level 1.

6.2. Level Start Scene:
Every monitor should display the same cut-in. The cut-in should display the current level, which needs to be relayed from the mother board.

6.3. Ingame Scene:
Each monitor should only display the aliens that are visible in it. An alien should at most be visible in two monitors.
Aliens move radially 15 degrees once per update clock cycle (which gets shorter as level increases), and towards the player 1 step per 6 (or more, as level increases. Should be a multiple of 6) clock cycles. On the 8^{th} step an alien takes, it is game over. Aliens can be defined polymorphically, and the exact radial movement patterns don’t necessarily have to be linear.

6.4. Game Over Scene:
The final level and score should be displayed at the top, then a three-character field to input a player tag with, and finally the scoreboard, which should

2. 規劃工作項目、進度與分工

Task 1: Establish synchronization protocols between the mother board and the child boards.
Comment: This is one of the greater technical hurdles of this project. This should be done at the very least and is alone a substantial technical achievement. (Estimated Time: 2 Weeks)

Task 2: Connect the Dance Pad to the mother board.
Comment: There may be some difficulty with figuring out how to interface the USB port. (Estimated Time: 1 Week)

Task 3: Set up VGA connections for each board.
Comment: If you can do it for one board, the other three boards can be done in the same way. (Estimated Time: < 1 Day)

Task 4: Set up audio connection for the mother board. Optionally, set up audio connection for the other three boards to achieve surround sound.
Comment: If you can do it for one board, the other three boards can be done in the same way. Task 1 to 4 can all be done in parallel development-wise. (Estimated Time: < 1 Day)

Task 5: Preprocess game resources. (ie. Sprites, SFX)
Comment: This should be standardized and is likely going to be covered in homework. (Estimated Time: < 1 Day)

Task 6: Model game logic as an FSM in the mother board and perform trivial parallelization whenever possible. Rendering is done in parallel, for instance. (Estimated Time: 2 to 4 Weeks)

Task 6.1: Consider the distance between the player and the monitors, and create false perspective using grid transformations to emulate spherical projection.

Task 6.2: Implement difficulty scaling for every "level" cleared. This is optional, but also trivial. The simplest difficulty scaling is to increase the movement speed of the aliens, after all.

Task 6.3: Implement a scoreboard of size 5. A sorting network may need to be implemented (this can be tested entirely in simulation). Optionally, set up keyboard connection (or use a cyclic alphabet input interface, a classic FSM design) to accommodate three-digit name input. If there is no time, at the very least the final score should be displayed.

Comment: The other technical hurdle. Parallelization is limited to trivial cases to avoid having to deal with data races, which are infamously difficult to debug in software, let alone hardware. The good news is that there is basically no physics emulation involved, which would significantly reduce the number of possible bugs and accelerate development time.

Task 7: Implement a credit interface to attach to the mother board using a pushbutton to emulate the interface. (Estimated Time: < 1 Week)
Comment: The finishing touch.

3. 可能遭遇之困難與預期解決方法或備案

I suspect that once the project gets into game logic, all the data coordination problems will arise, so Task 6 will be the chokepoint of the project. Luckily, a video game is conceptually flexible, and when we can't overcome a certain technical difficulty, we can transform the game to bypass that difficulty.

If it comes down to the worst case – that we can't get the synchronization protocols

working, then honestly the project has lost its technical point. So, we will put extra effort into that part, even if it takes up more than half the development time. (Though the projected goal is to finish it in 2 weeks)