Design Document #4 for Mini Project – Currency Dispenser

1. Detailed Description of the Overall Design

Our project is a currency dispenser which administers monetary value to the user according to the given input specifications.
The money dispenser has notes of the following denominations.

Acronym	**Denomination**	**No.**
0n	1	99
fi	5	50
te	10	99
hu	100	99

The user selects the required mode through 2 switches at the right bottom corner. The default mode is the **Display Mode (00)** which is used to exhibit the number of notes currently in the repository. This is done through the seven segment display which specifies the acronym for the denomination followed by the number of notes left. The user needs to select the denomination (that will be operated upon) through the third and fourth switches from the right.

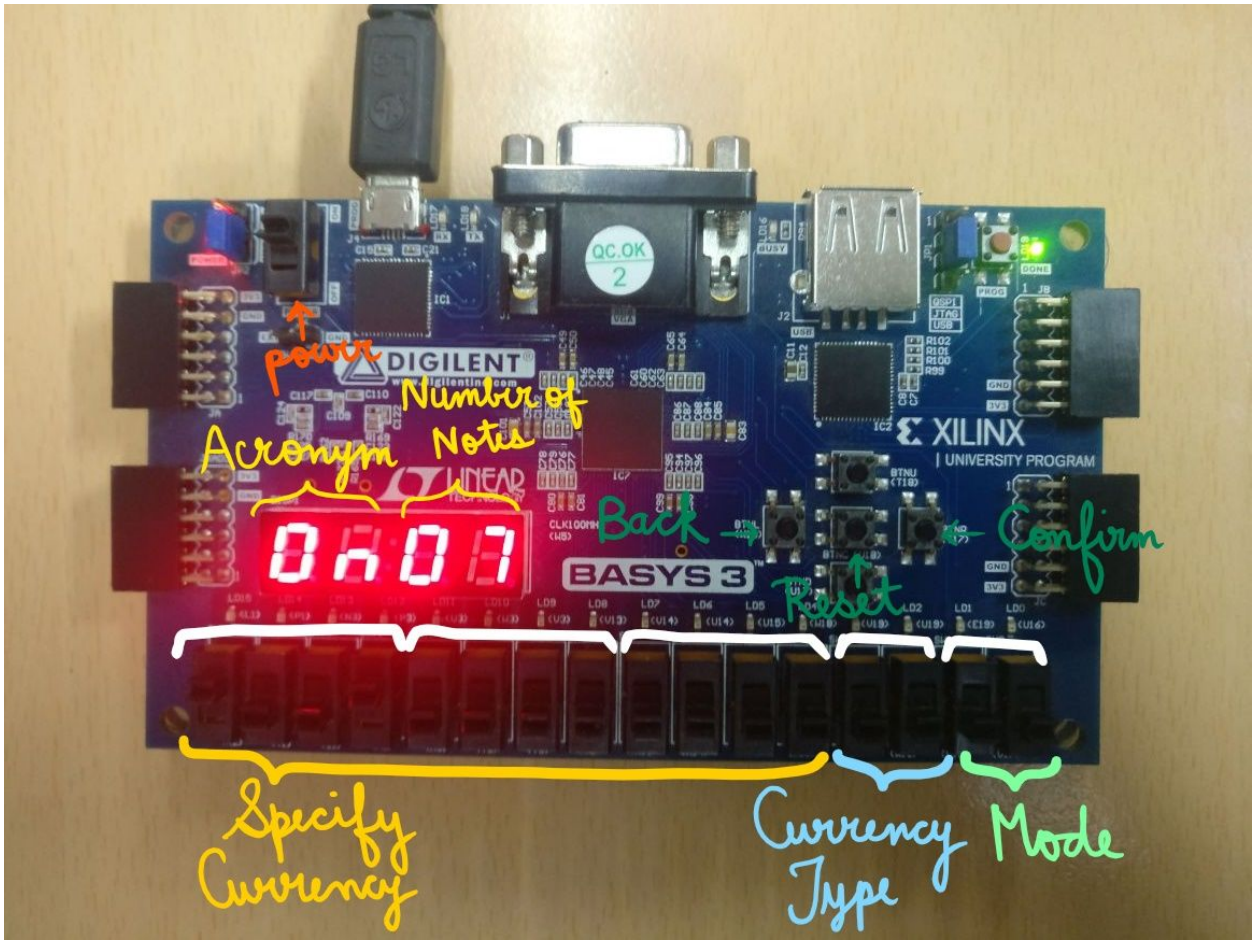
- 00 - One
- 01 - Five
- 10 - Ten
- 11 - Hundred

We offer three input modes:

1. **Specify Mode (11)** : The user specifies the requirement through giving the choice of denomination in the same way as display mode and the number of notes for withdrawal through first eight switches on the board from the left side. The first four switches correspond to the first digit and the second set of four switches, to the second digit. For example, to withdraw 83 ten rupee notes, the user will need to set 10 on the denomination selection switches i.e. the third and fourth ones from the right and set 1000 0011 on the switches starting from the left 1000 for first digit 8 and 0011 for the second digit 3. When done setting the switches, the user must press the confirm button (push button on the right side). If the required notes are not present in the machine, dispenser will show an error message. The machine can be returned to the previous

state by the back button (push button on the left side) or resetted to the original state by the reset button (push button in the middle).

2. **Highest Denomination Mode (10)** : Dispenser will generate the highest denominations present in the dispenser for the amount specified by user. It will throw an error if the specified amount cannot be transferred using the highest denomination notes available. For example, if the user asks for 340 rupees, the dispenser will give 3 hundred and 4 ten rupee notes. If they are not available, the algorithm throws an error.
3. **Auto Mode (01)** : The user will specify the amount and dispenser will generate mixed notes. The mixed notes will be generated according to the monetary value asked for withdrawal. This the mode in which bank ATMs function because the algorithm tries to minimise the number of notes to be dispensed. It gives priority to higher denomination notes but if they are not available, it will try to suffice by dispensing lower denomination notes according to the availability. It will throw an error only if it is impossible to provide the asked amount through all possible permutations of the total available notes in the bank. For example, if the user asks for 768 rupees and the bank has only 5 hundred, 12 ten, 19 five and 77 one rupee notes available, it will dispense the required amount using all hundred, ten and five notes in addition to using 53 one rupee notes. The bank will only have 24 one rupee notes left after this transaction.



2. Description of Major Components, Subsystems and Interconnecting Signals

The major components are the display block and the main entity working on different modes. We have tried to break the code into clear logical blocks. The main entity is broken about the different mode values using an overarching if block. The different mode values are specified above and each mode has been executed **independently** of each other.

1. **Mode** :- This is the input which defines the specific mode on which the currency dispenser will work.
2. **Err** :- In case of a wrong removal of money in highest denomination or auto mode an error is returned. This would trigger the LED by passing it to the display block.
3. **Show_currency** :- This is a connecting signal between the display and the main entity. This would specify the number that should be shown on the seven segment display. It is shown in Binary Coded Decimal.
4. **Show_type** :- This is another connecting signal which specifies the type of note that must be shown on the SSD. It also shows the abbreviation of the type to be shown.
5. **Money_asked** :- This integer holds the money asked from the dispenser converted from its binary coded decimal.
6. **Notes_100**, **Notes_10**, **Notes_5**, **Notes_1** :- These are registers that hold the amounts of different notes that are present in the dispenser at any given time.
7. **Back** :- In case of an error, no further computation takes place unless this push-button is pressed.
8. **Confirm** :- Pressed to trigger the transaction required. Will lead to the error if availability issues prevail.

3. Testing Process

We have validated our program using the following methods:

- **Unit Level Testing of subsystems:**
    a. **Dispenser Entity using Testbench file**: We designed a testbench file for testing our dispenser unit which works in four different modes according to user specified input. The testbench tests a total of 22 varied cases that are sub-sectioned in the form of two sequential operations list. We have specified possible error messages and comments at each step for easier debugging in case of error. We successfully tested the testbench on the dispenser entity without any error.
    b. **Display Entity using FPGA Board**: We tested our display block on the board itself by specifying various inputs and displaying the output. It matched with the

desired requirements and we were successfully able to test our display block without any error.

- **Top Level Testing on the FPGA Board:** We tested our entire program by generating a bitstream and programming the FPGA board. We tested various test cases in a sequential fashion. These consisted of successful monetary transactions from the dispenser in various currency modes interspersed with cases that would generate errors. Hence, covering all possible combinations that can occur when used by consumers in a real life application scenario. In case of error, the back button is used to go the previous state and the process of successful transactions can continue. The dispenser is replenished using the reset button.

Example of sequential operations sets:

- Testbench Sequential Steps Set-1
    - 1. Specify Mode - hu: 94, te: 89, fi: 40, on: 49
    - 2. Highest Denomination Mode - 700
    - 3. Auto Mode - 702, 700

  This set basically withdraws 94 hundred, 89 ten, 40 five and 49 one rupee notes in specify currency mode. This leaves 5 hundred, 10 ten, 10 five and 50 one rupee notes in the bank which totals up to 700 rupees. Next 700 rupees is asked to be withdrawn in Highest denomination mode which results in an error due unavailability of 7 hundred rupee notes. This is followed by a call to withdraw 702 rupees which exceeds the total bank money resulting in yet another error. Then finally 700 rupees are withdrawn in Auto mode successfully ending the sequential step checking process.
- Testbench Sequential Steps Set-2
    - 1. Highest Denomination Mode - 345
    - 2. Specify Mode - hu: 96, te: 95, fi: 50, on: 00
    - 3. Auto Mode - 344

  This set successfully withdraws 345 rupees in Highest denomination mode followed by specific withdrawals in specify mode. This results in 0 hundred and 0 ten rupee notes. Next withdraw tries to take out 50 five rupee notes resulting in an error. Finally, 344 rupees are withdrawn successfully ending the sequential step checking process.