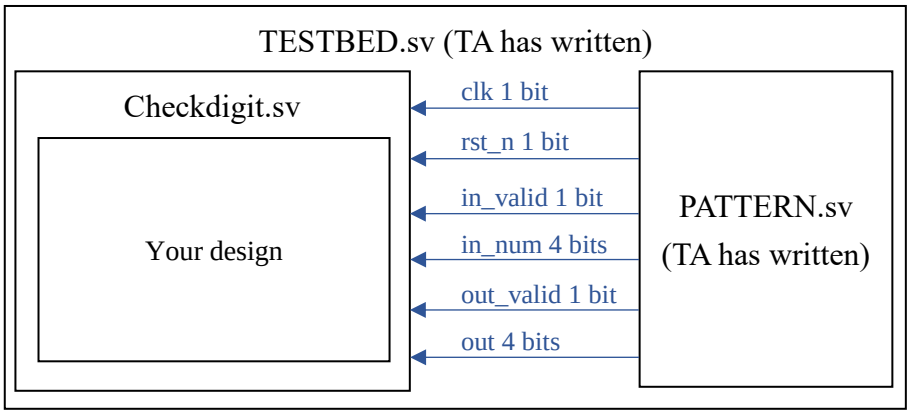
# NYCU-ECE DCS-2022

HW02　Design: Visa Check Digit

## 資料準備

1. 從 TA 目錄資料夾解壓縮:
   **% tar -xvf ~dcsta01/ HW02.tar**
2. 解壓縮資料夾 HW02 包含以下:
   a. 00_TESTBED/
   b. 01_RTL/
   c. 02_SYN/
   d. 03_GATE/
   e. 09_UPLOAD/

## Block Diagram




## 設計描述

Check digit is the last digit in a number sequence. One of its applications is in banking system to protect us against inadvertent typo. (e.g. bank account number, credit card number)

**急了！匯錢匯錯帳戶急聯絡對方「淡水陳先生」不接電話也不回訊**

5

陳彩梅

2022年2月23日 · 3 分鐘 (閱讀時間)

如今ATM或線上轉帳匯款都相當方便，不必先把現金領出來，就能直接進行交易，不過在匯款之前，最好多檢查匯入帳號是否正確，以免不小心匯錯款。近日就有一名網友無奈表示，匯款給家人時，不小心按錯一個號碼，結果把錢轉進陌生人的帳戶，雖然銀行在取得對方同意後，把電話號碼留給原PO，沒想到對方卻不接電話，傳LINE也不回，讓原PO相當焦急。貼文一出，隨即掀起網友熱議。

Each visa credit card has 16 digits, and the last digit is the check digit. In this homework, you will receive the first 15 digits of a visa credit card. You have to calculate the last digit, i.e., the check digit according to the rule. Refer to the figure below for detailed rule and example.

(Rules from different banks may vary. We use the rule from firstbank)

信用卡號檢核公式

- **權數為固定值。**

卡片號碼（Card Number）（長度：16 ）

發卡單位 BIN：6 碼

卡片流水號　：7 碼

正附卡序號　：1 碼

補發卡序號　：1 碼

卡片檢查碼　：1 碼，計算公式如下：

範例 1：VISA 卡片號碼：456303-0100570-3-0-6

| | 4 | 5 | 6 | 3 | 0 | 3－0 | 1 | 0 | 0 | 5 | 7 | 0－3－0 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 權數 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 |
| 相乘值 | 8 | 5 | 12 | 3 | 0 | 3 | 0 | 1 | 0 | 0 | 10 | 7 | 0 | 3 | 0 |

將上述各項值相加（若含２碼，例如１２，則拆為１與２）

８＋５＋１＋２＋３＋０＋３＋０＋１＋０＋０＋１＋０＋７＋０＋３＋０＝３４

３４取個位數４，以１０減４即得檢查碼６。

The output will depend on the validity of this credit card. There are two cases:

1. The credit card number is valid if your answer is one digit (1-9), pull out_valid high and output your answer.
2. The credit card number is not valid if your answer is two digits, pull out_valid high and output ”15” to indicate a violation.

## Inputs

| Signal name | Number of bit | Description |
|---|---|---|
| clk | 1 | 10ns clock signal |
| rst_n | 1 | Asynchronous negedge reset signal |
| in_valid | 1 | Pulled high during digit inputs |
| in_num | 4 | Random 4-bit credit card digit (0-9) |

## Outputs

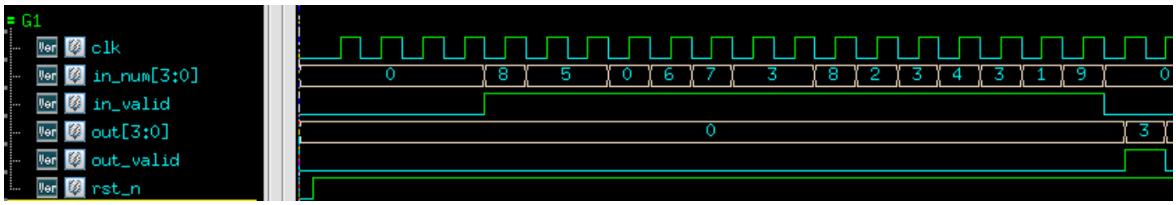
| Signal name | Number of bit | Description |
|---|---|---|
| out_valid | 1 | Pulled high during check digit output<br>(reset required) |
| out | 4 | Check digit output<br>(1-9 for valid card, 15 for not valid card)<br>(reset required) |

## Specifications

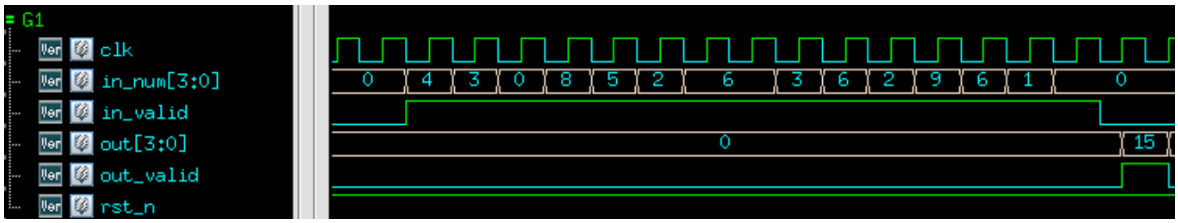
1. Top module name: **Checkdigit** (File name : **Checkdigit.sv**)
2. 在非同步負準位 reset 後，所有的 output 訊號必須全部歸零。
3. Output 要在 Input 結束後的 100 cycles 內輸出。
4. Output 僅能輸出 1 cycle，不能多不能少。
5. 02_SYN result 不行有 error 且不能有任何 latch。
6. Clock period 10 ns。
7. Input delay = 0.5 * clock period; Output delay = 0.5 * clock period
8. Separate your combinational and sequential blocks!

## Example waveform

### Case1: a valid card number, check digit is 3




### Case2: a not valid card number, output should be 15




## 上傳檔案

1. Code使用09_upload上傳。
2. report_dcsxx.pdf, xx is your server account. 上傳至new E3。
3. 請在 3/31 (四) 15:30 上課之前上傳

## Grading policy

1. Pass the RTL& Synthesis simulation. 70%
2. Performance 20%
   Ranking formula: (total latency + number of patterns) * area
3. Report 10%

## Note

Template folders and reference commands:

1. 01_RTL/ (RTL simulation) → **./01_run**
2. 02_SYN/ (synthesis) → **./01_run_dc**
3. 03_GATE/ (gate-level simulation) → **./01_run**
4. 09_UPLOAD/ (upload) → **./09_upload**

報告請簡單且重點撰寫，不超過兩頁A4，並包括以下內容

1. 描述你的設計方法，包含但不限於如何加速(減少critical path)或降低面積。
2. 基於以上，畫出你的架構圖(Block diagram)
3. 心得報告，不侷限於此次作業，對於作業或上課內容都可以寫下。
4. 遇到的困難與如何解決。