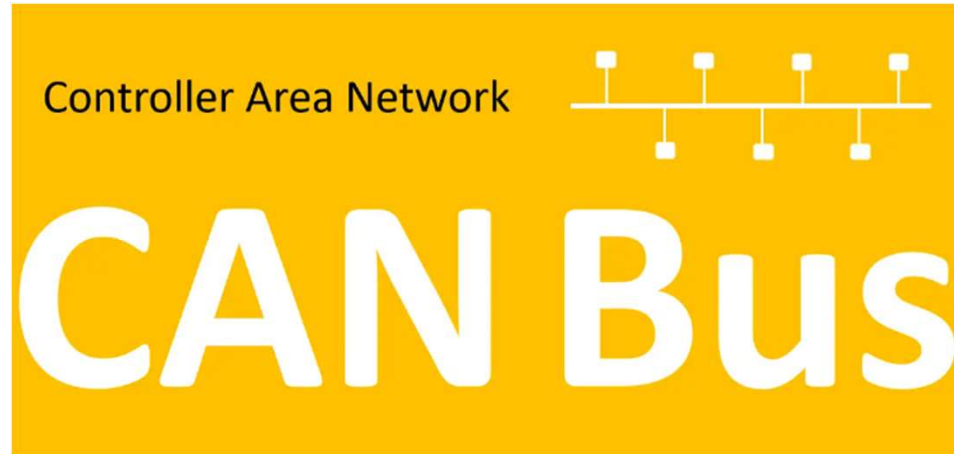


# 使用FPGA設計— CAN BUS控制器與其應用。

王鴻傑 2023.11/15

9:30-11:30



CAN 2.0A

CAN 2.0B

CAN FD

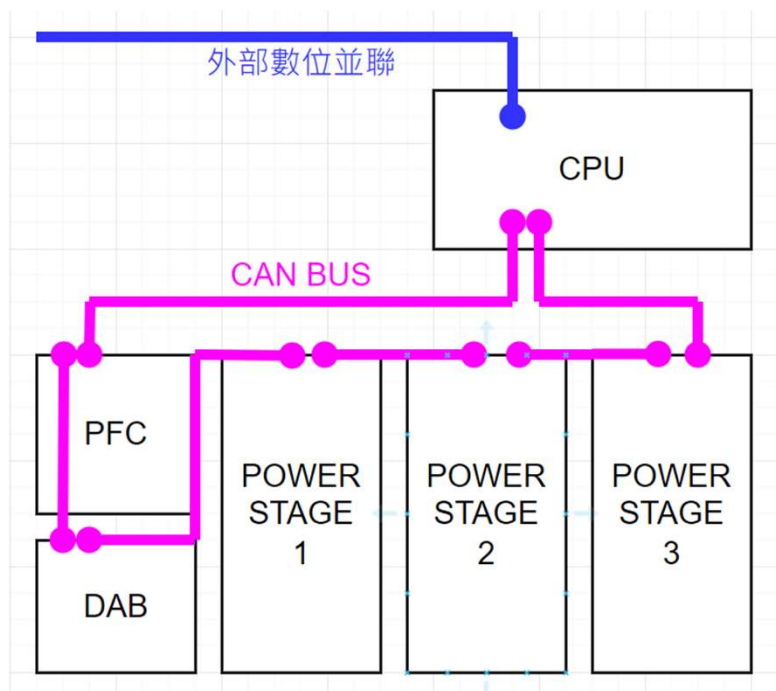
# 動機

1. 在通訊BUS上可新增或刪除通訊節點(Node) 。
2. 具備優先權仲裁，CRC，ERROR，IDEL/ASYNC等...
3. 主機命令可以具有 “同時性” 。
4. CAN控制器+自動管理(數位電路) 。

# 目標

1. 1Mbps(固定)。
2. CAN Basic (不是CAN Extended, 不是CAN FD)。
3. 事先分配ID的功能。
4. 分配一個BRAM做為控制器與CPU的資料同步
5. 提供與Modbus相似的三個命令。
  - 寫入一個Byte資料。
  - 寫入一區塊(連續)資料。
  - 讀取一區塊(連續)資料。

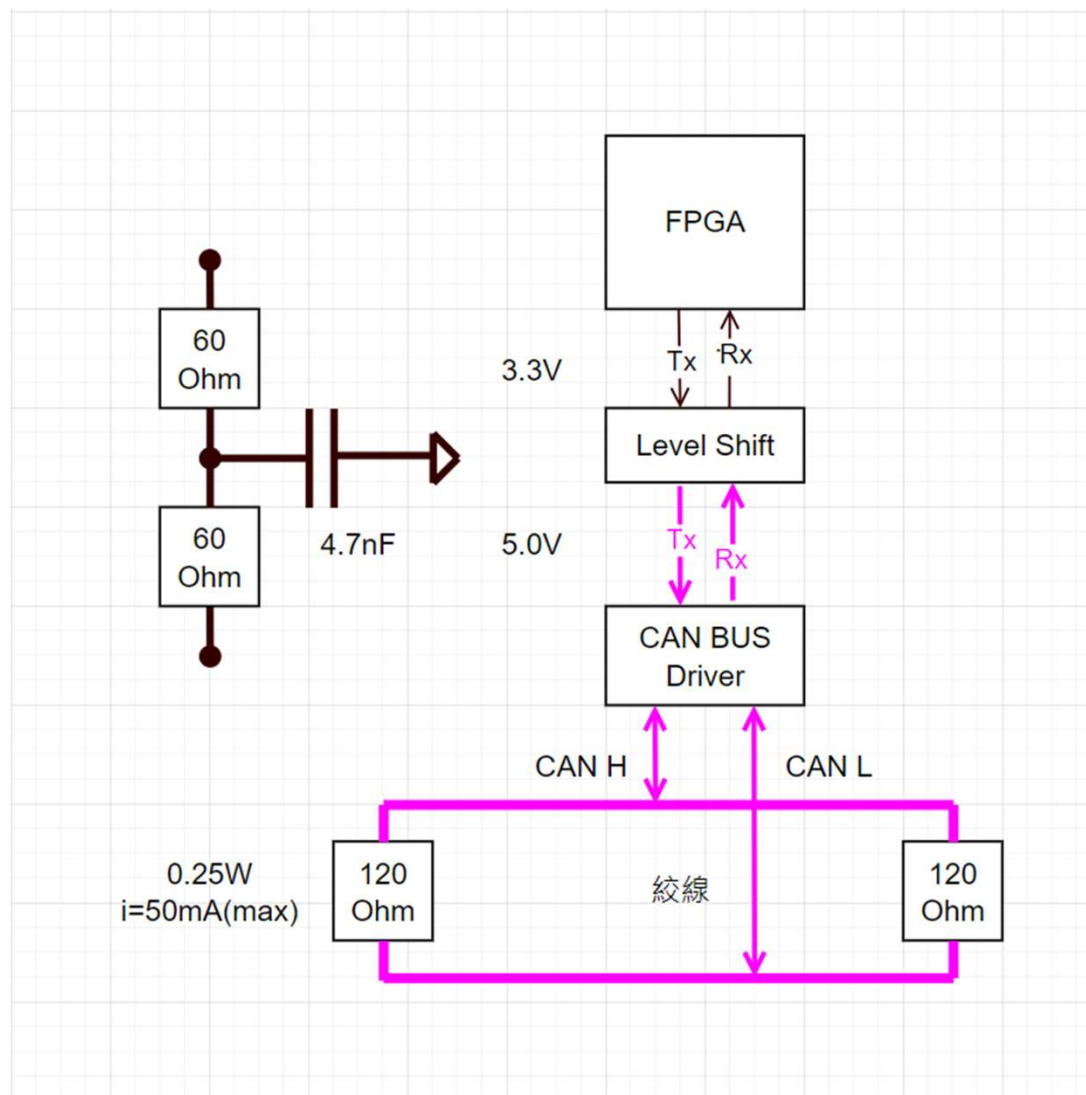
# 運用



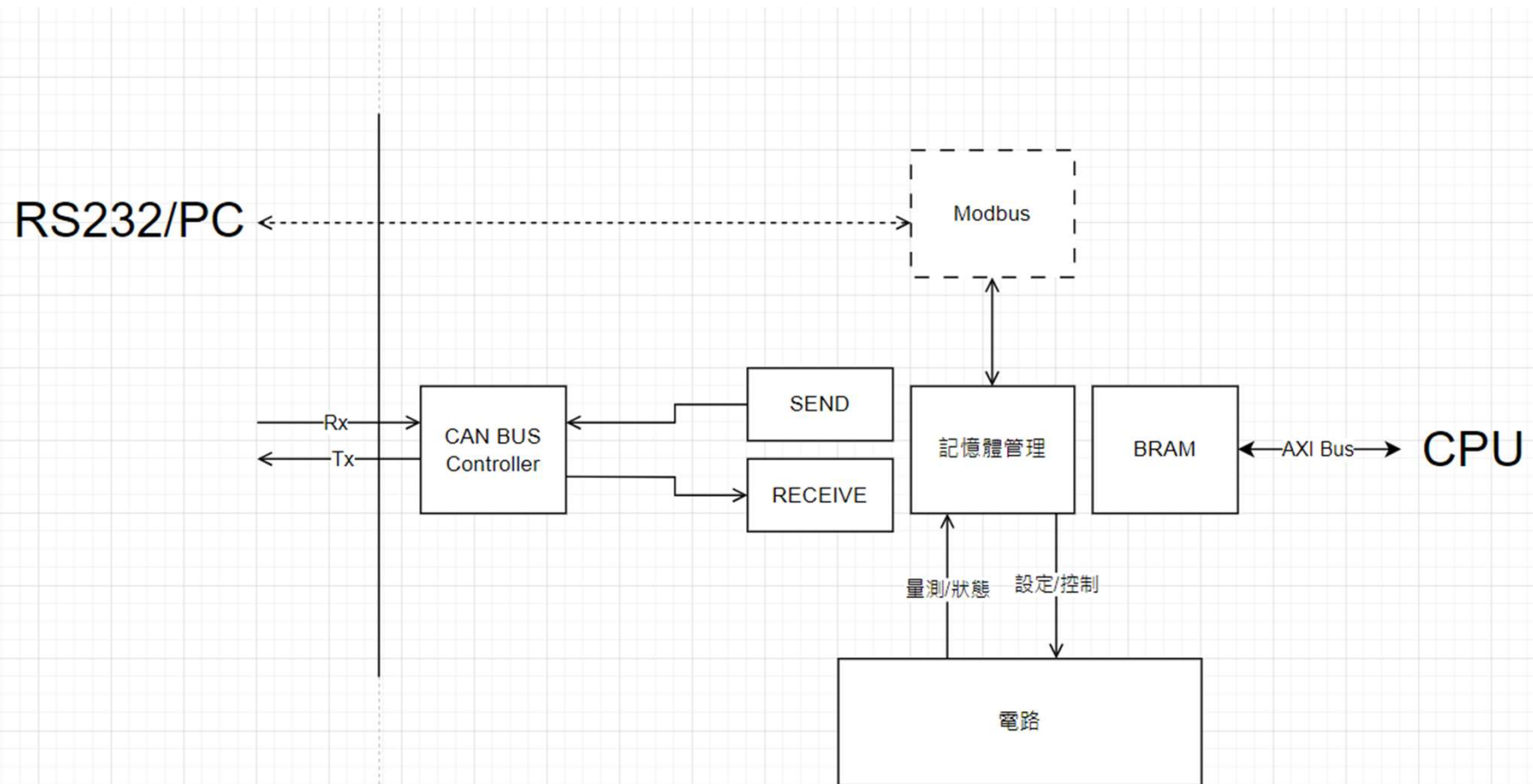
## Memory Map for CAN(Internal use)

Node Name	Address	Function
ID[10:7]	ID[6:0]	
0	X	<RESERVE>
1	X	<RESERVE>
2	127-0	CPU
3	127-0	POWER STAGE I
4	127-0	POWER STAGE II
5	127-0	POWER STAGE III
6	127-0	PFC
7	127-0	DAB
8	X	<RESERVE>
9	X	<RESERVE>
A	X	<RESERVE>
B	X	<RESERVE>
C	X	<RESERVE>
D	X	<RESERVE>
E	X	<RESERVE>
F	X	<RESERVE>

# CAN DRIVER IC與FPGA連接



# FPGA 設計概念



Controller Area Network



# CAN Bus



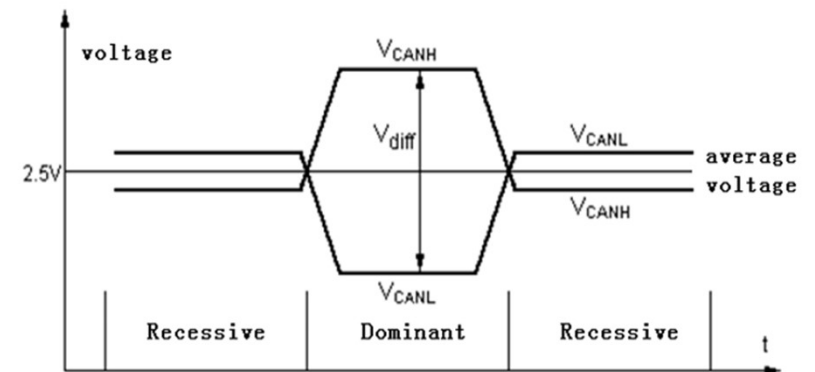
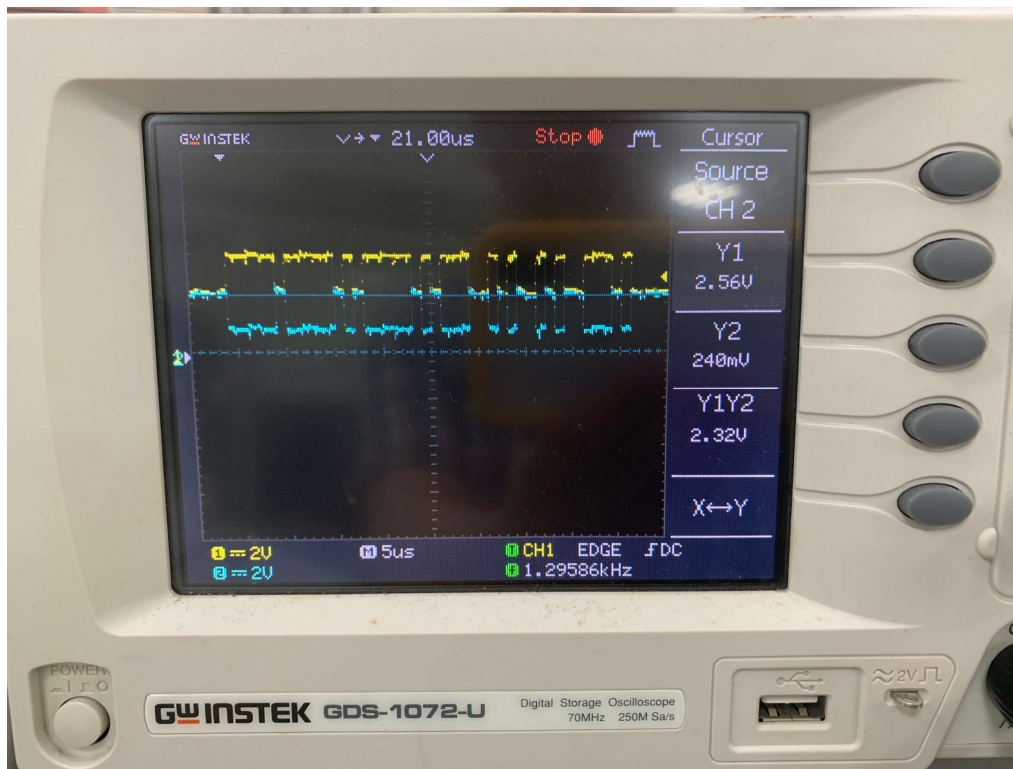
# 要了解，才能設計。

1. CAN BUS Electric Signal
2. Message type
3. Error Type
4. Error Manager
5. 傳送時機? (Bus Idle)
6. Bit timing
7. BUS Arbitration
8. Bit Monitor, Coverage, BIT Error
9. Bit stuffing, Coverage
10. CRC 15, Coverage
11. ACK response, Coverage
12. De-stuffed / Stuffed Error
13. CRC Error
14. FORM Error
15. ACK Error
16. Error Manager
17. ID Mask or Filter

# 示波器上看到的訊號

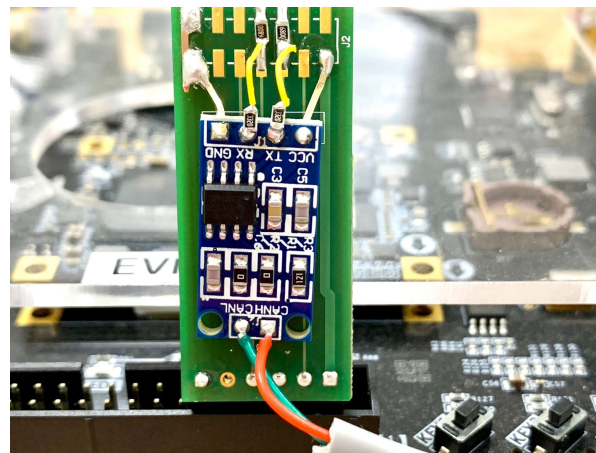
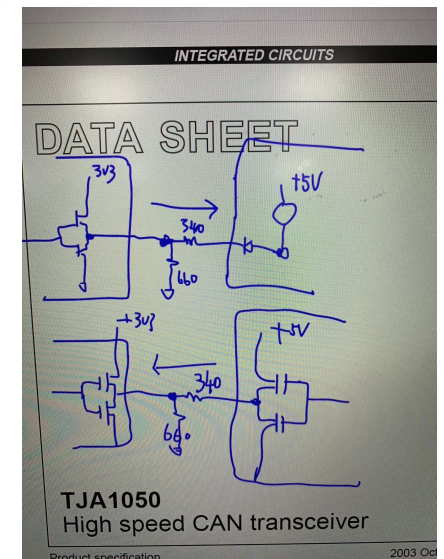
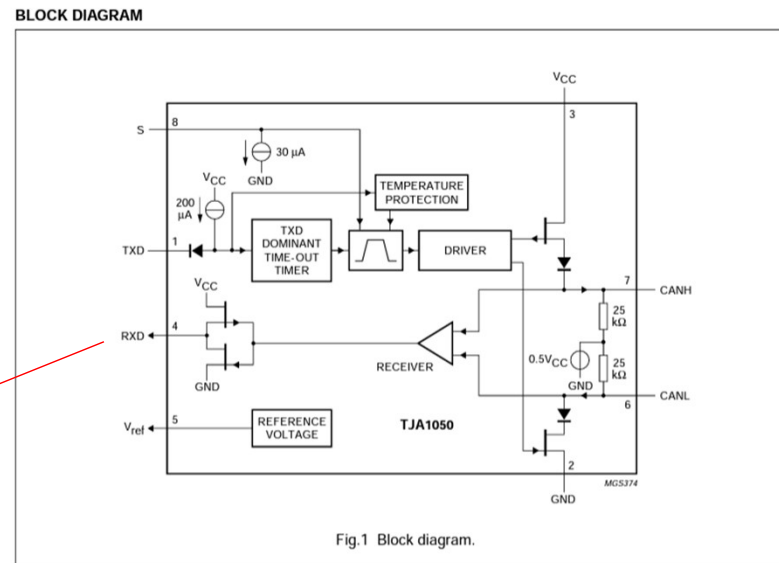
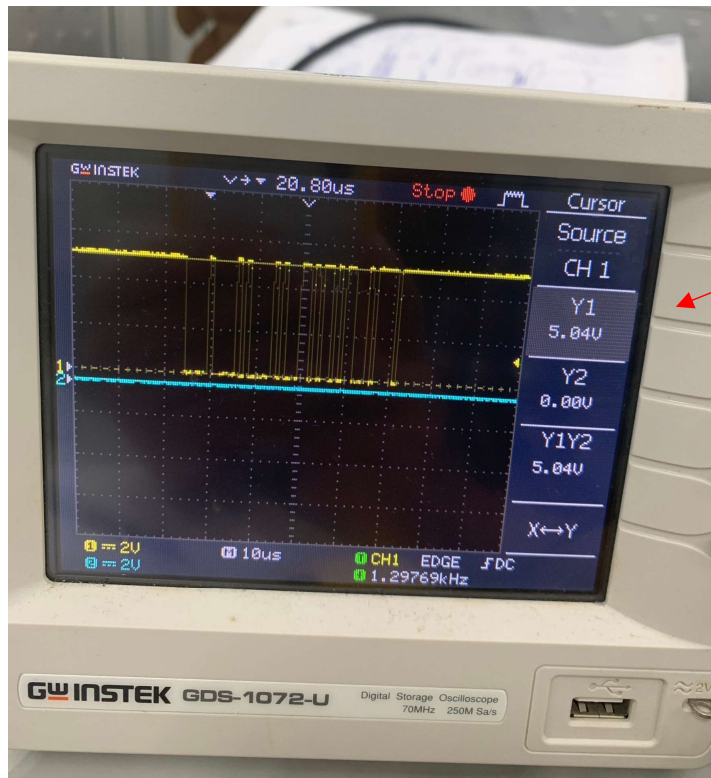
Physical Layer :

1. Dominant(0) & Recessive(1).
2. Dominant 可以覆蓋 Recessive



# FPGA上看到的訊號

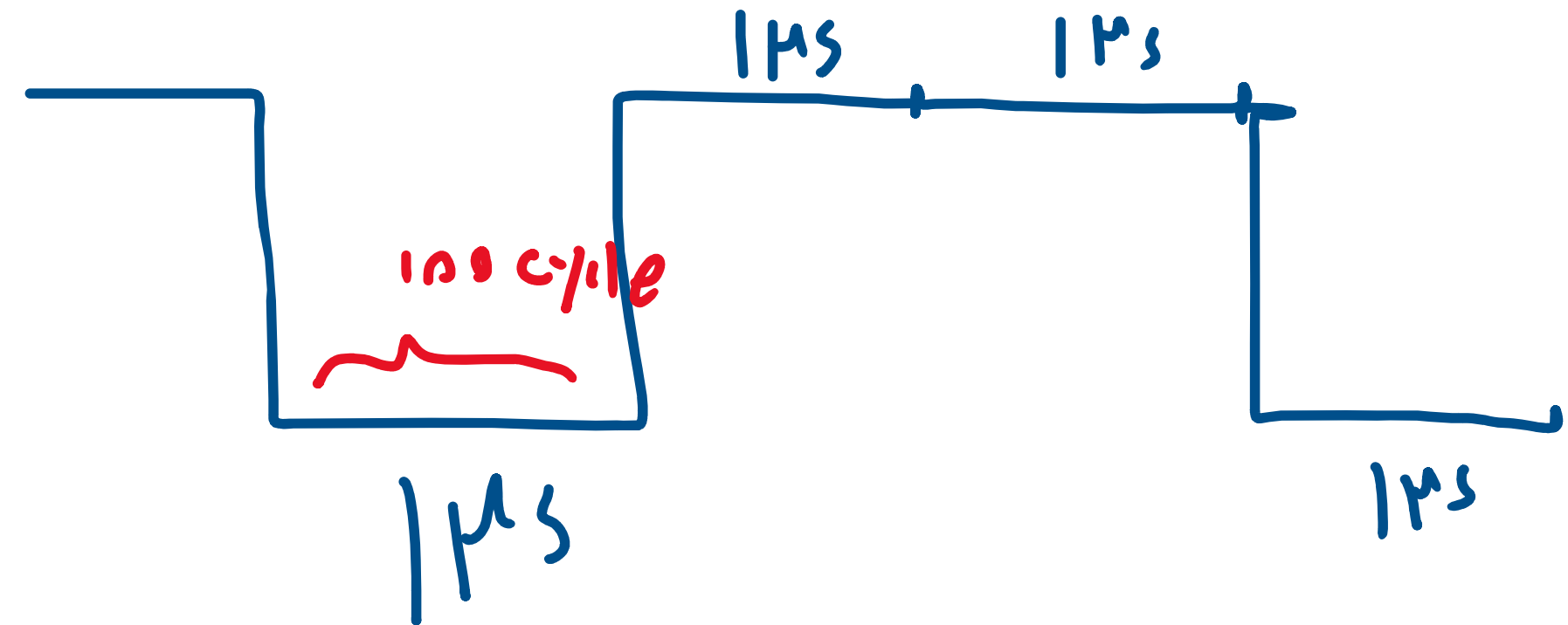
Idle: logic 1



Level Transfer

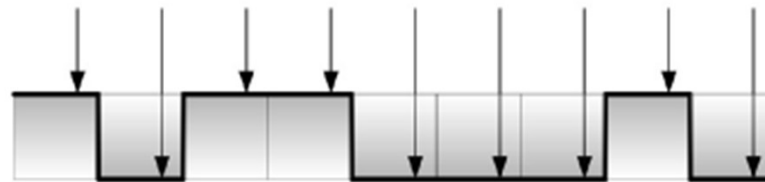
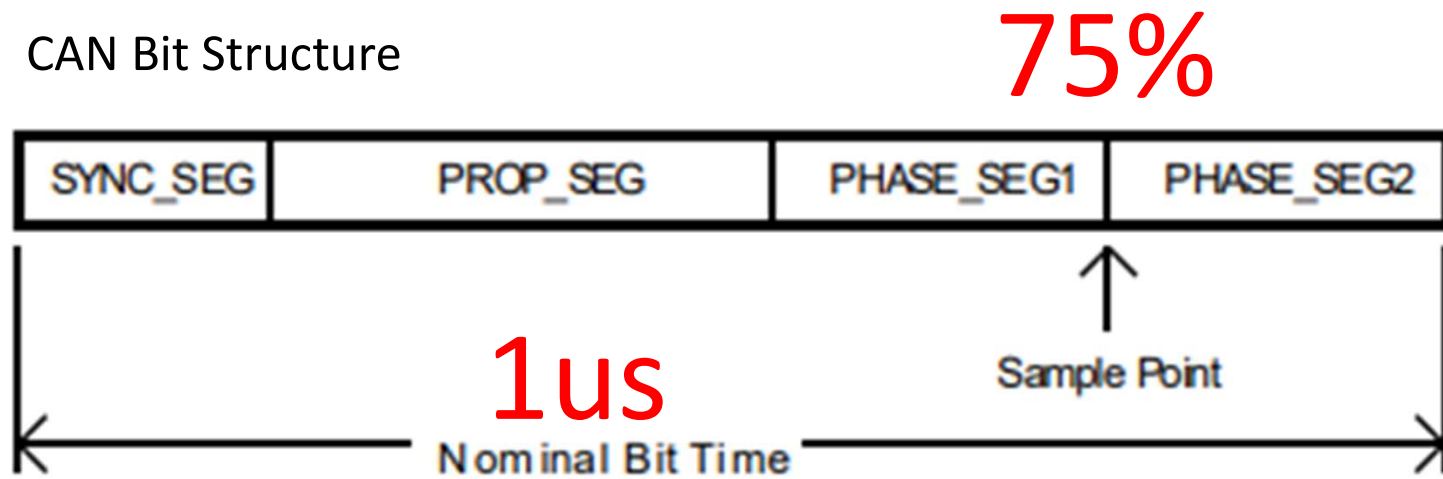
# CAN Bit Rate

```
parameter CLK_HZ = 100000000; // 100MHz  
parameter BITRATE_HZ = 1000000; // 1Mbps  
parameter BIT_CYCLE = (CLK_HZ/BITRATE_HZ);  
parameter SMAPLING = (BIT_CYCLE *75)/100; // Sampling @ 75% bit
```



# Bit 取樣點

CAN Bit Structure

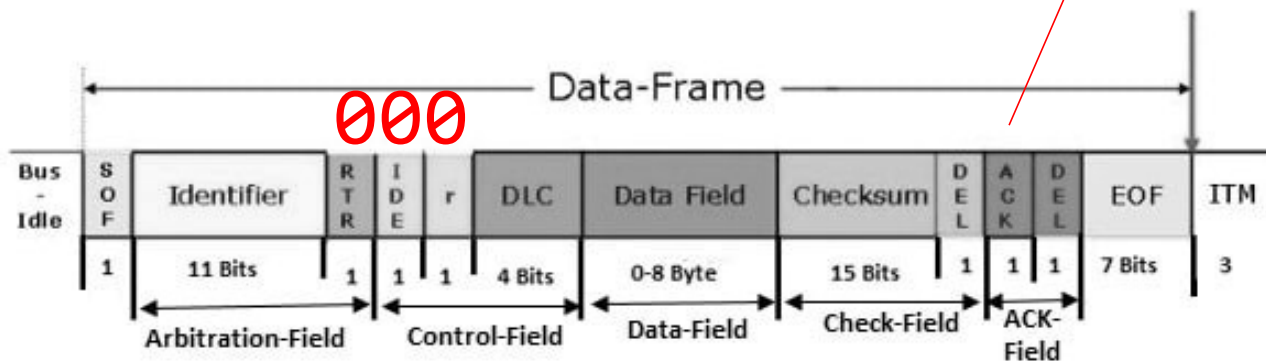


# Frame種類

## 1. Data Frame

包含節點資料的

檢查Rx



RTR, Remote Transmission Request。Data(0)/Remote(1)

IDE, Standard ID(0)/Extended ID(1)

r, Reserved bit, 必須為Dominant(0)

ACK(slot), 資料都是送出Recessive。

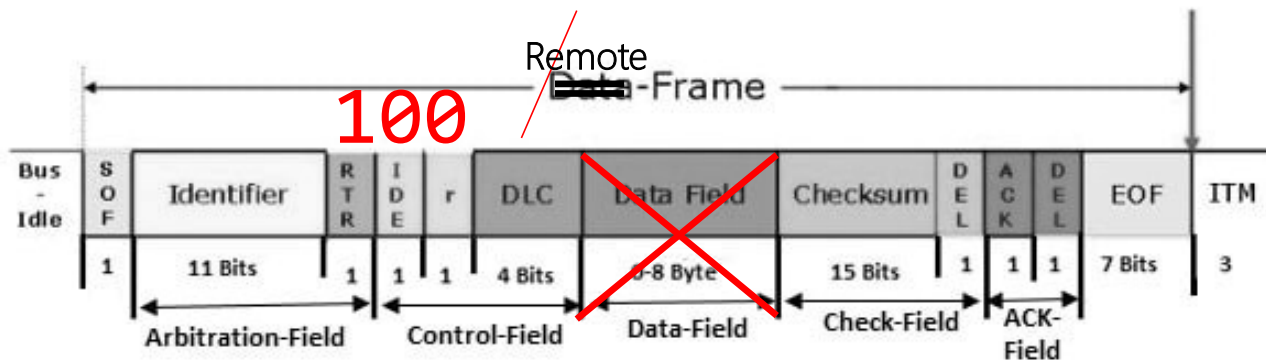
若BUS有節點接收成功，會將ACK改為Dominant。

# Frame種類

## 2. Remote Frame

取得某個ID的資料

要填入欲接收資料長度



RTR, Remote Transmission Request。Data(0)/Remote(1)

IDE, Standard ID(0)/Extended ID(1)

r, Reserved bit, 必須為Dominant(0)

ACK(slot), 資料都是送出Recessive。

若BUS有節點接收成功，會將ACK改為Dominant。



# Frame種類

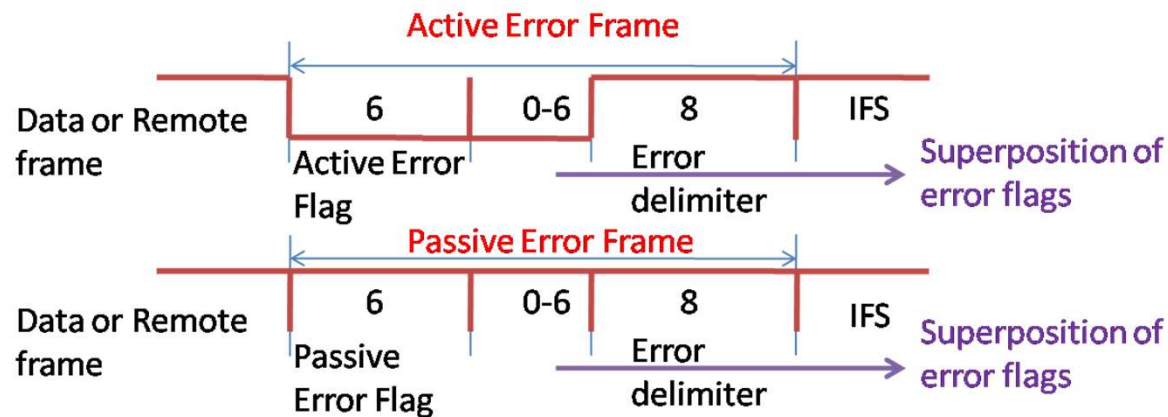
## 3. Error Frame

兩個部分組成。錯誤旗標(Error Flag)，錯誤邊界(Error Delimiter)

主動錯誤旗標(Error Active): 連續6個Dominant Bit

被動錯誤旗標(Error Passive): 連續6個Recessive Bit

錯誤邊界: 連續8個Recessive Bit





# 對於Error的處理

為了正確傳遞資訊，CAN通訊嚴格定義了許多的錯誤狀態。

1. 錯誤偵測：檢測訊息封包細節，若有錯誤就發送Error Frame通知節點**重傳**。
2. 錯誤頻繁節點排除: 節點發現自己經常錯誤，減少或禁止使用網路。  
(Fault Confinement)

# 錯誤狀態遷移

每個Node都有Transmission error counter與Receiver error counter。

當偵測到錯誤時，對應的counter就 +1。

當完成一次傳輸，對應的counter就 -1。

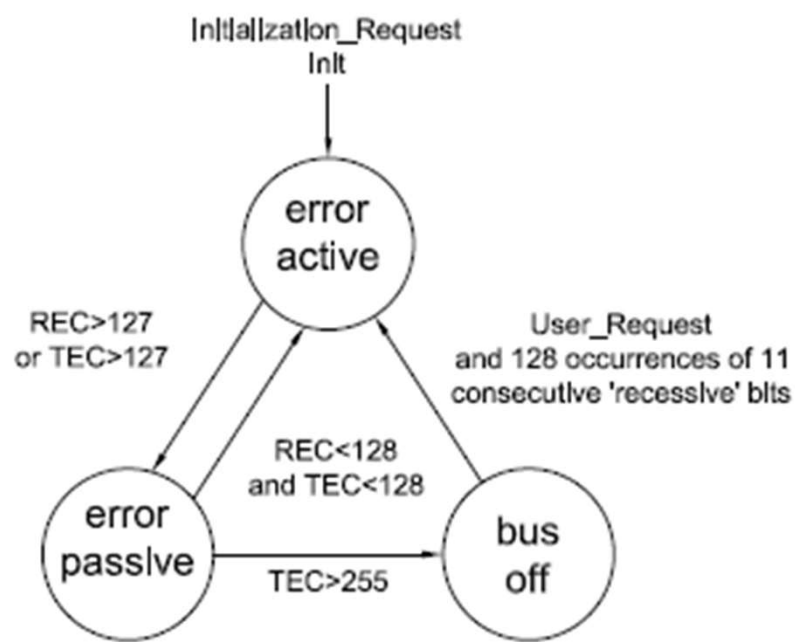


Figure 16 — Node status transition diagram

# 對於Error的排除

節點處在一種狀態。Error Active/Error Passive/ Bus Off

Error Active

偵測到錯誤就發送主動式錯誤旗標。

Error passive

偵測到錯誤就發送被動式錯誤旗標。此外在傳送間隔訊框需額外傳送8個 Recessive bit (Suspend transmission)。  
藉由暫停傳輸，其他Node才可以進行傳輸。

Bus Off

不參與網路通訊。

這個Node不會對網路造成任何影響。

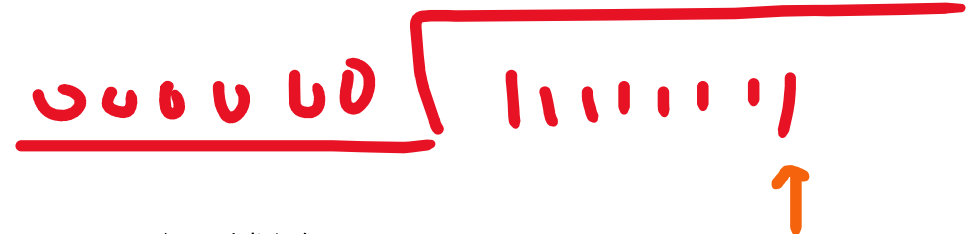
# 可能的錯誤來源

傳輸線故障，  
雜訊干擾太大，  
終端電阻不匹配，  
CAN Node故障。  
網路傳輸延遲(Network-induced delay)  
時脈不同步 (timer jitter)

...

# Frame種類

## 4. Overload Frame



兩個部分組成。過載旗標(Overload Flag)，過載邊界(Overload Delimiter)  
: 連續6個Dominant Bit組成過載旗標，連續8個Recessive Bit組成過載邊界。

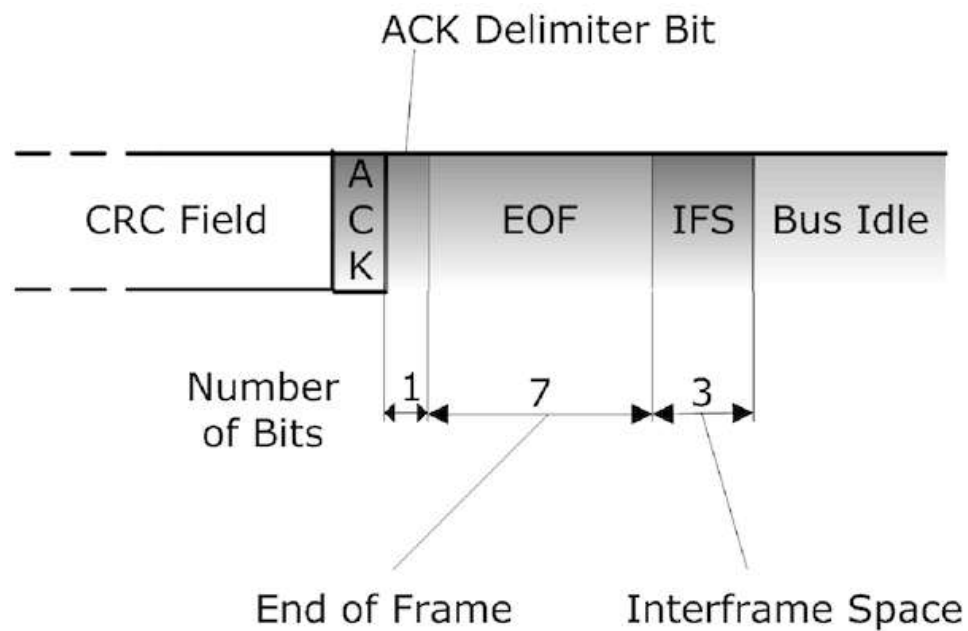
何時?發出

- 1 接收Node爭取處理資料時間。
- 2 Node在Interframe期間，偵測到 Dominant Bit。
- 3 Node在Overload Delimiter，Error Delimiter的第8個位元，偵測到 Dominant Bit。

# Frame種類

## 5. Interframe space

Data Frame與Remote Frame之間需要間隔欄框

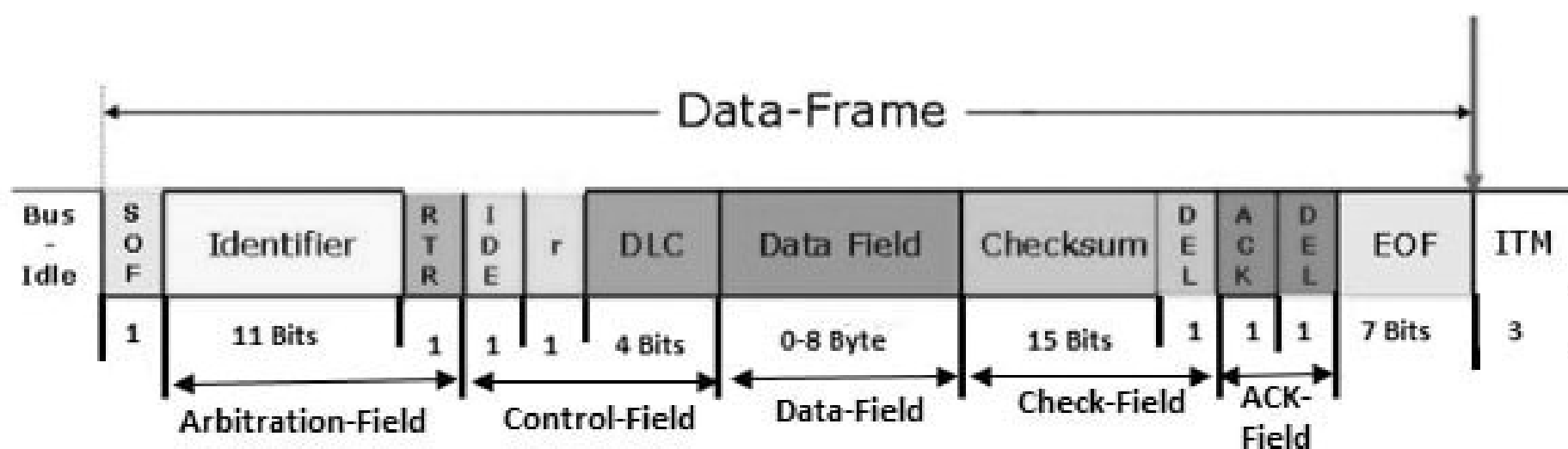


Controller Area Network



# CAN Bus

# 傳送與接收





# Bit-Stuffing

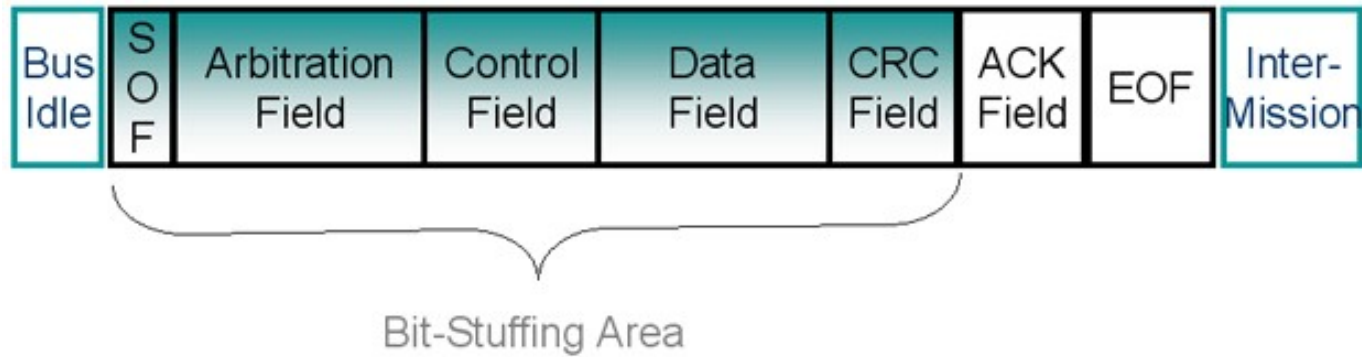
連續5個Dominant下個bit會填充1 bit Recessive。

連續5個Recessive下個bit會填充1 bit Dominant。

不會有6個連續的Dominant/Recessive

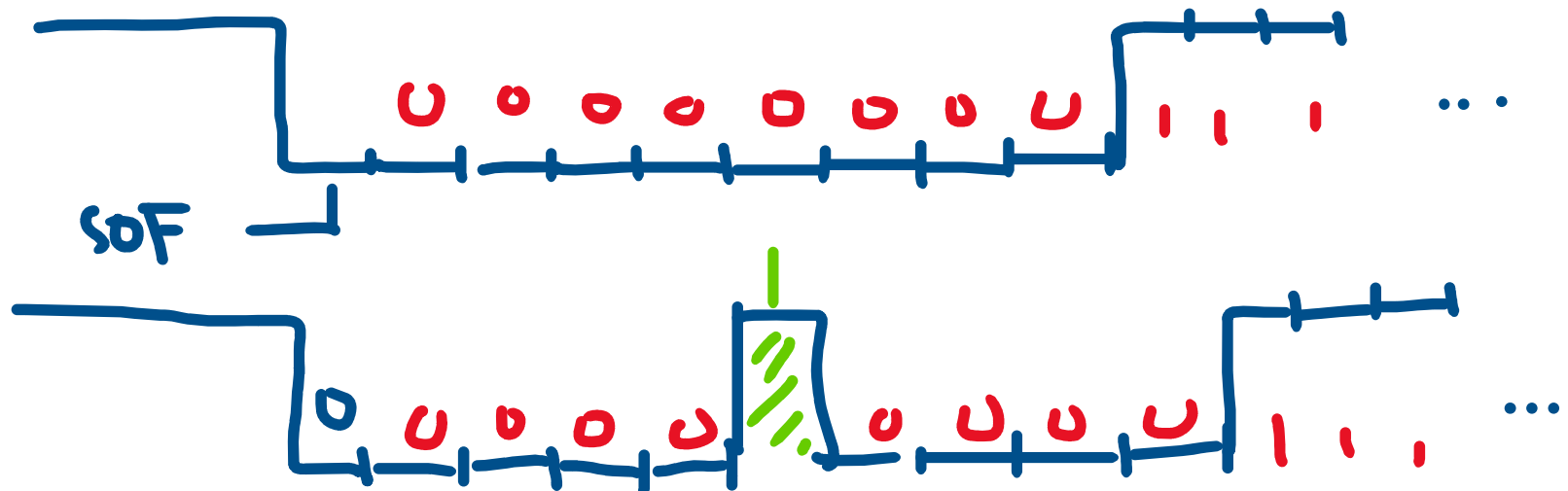
NRZ encoding

# Bit-Stuffing Area



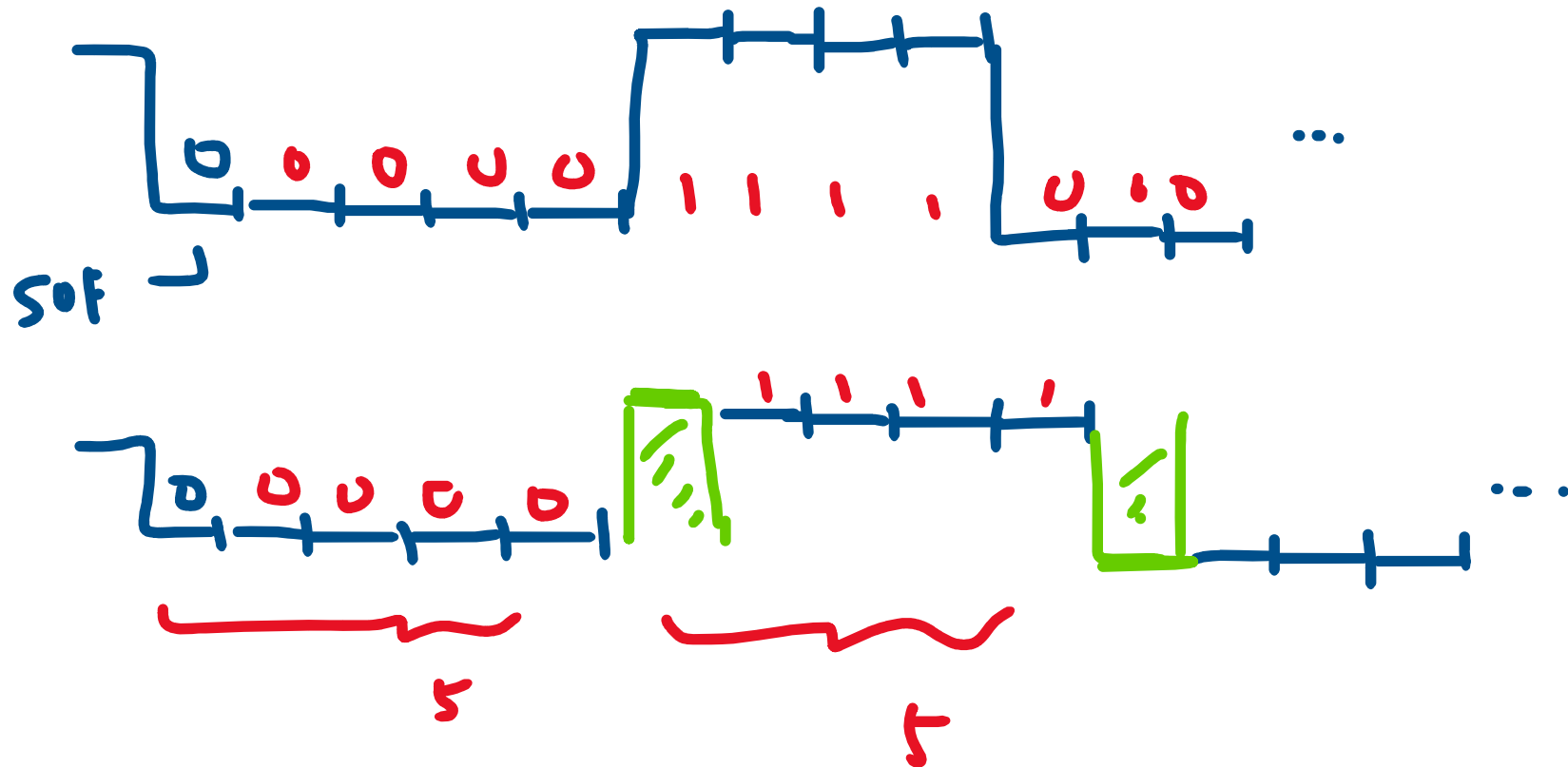
Example:

ID= 00000000111



Example:

ID= 00001111000



# Stuffing Error

所以在Stuffing Area中連續偵測到6個相同的bit的時候，就會從下一個bit開始傳送Error Frame。

# BUS Arbitration

ID數值越小，優先權位越高。

Dominant可以覆蓋Recessive

CAN BUS 屬於非破壞性仲裁機制，意指仲裁勝出者不須重新啟動或重傳，仲裁失敗者也不受影響。

**Example:**

**ID1= 00000000111**

**ID2= 00010000101**

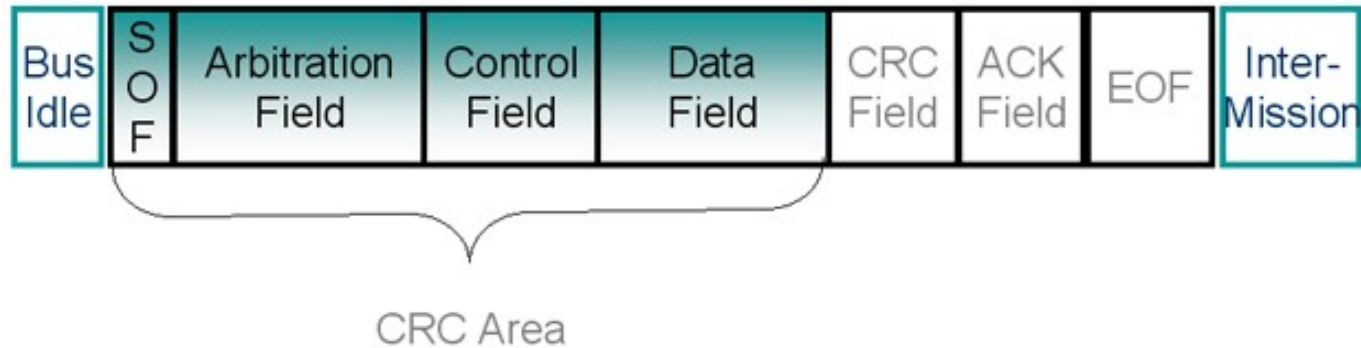
# CRC 15

In Classical CAN, a 15-bit CRC polynomial is used ( $x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + x^0$ ).

<https://www.ghsi.de/pages/subpages/Online%20CRC%20Calculation/indexDetails.php?Polynom=10100111010101011&Message=1>

```
1 module crc15(  
2     output reg [14:0]CRC,  
3     input  sdata,  
4     input  idle,  
5     //  
6     input  clk,  
7     input  sysrst  
8 );  
9     // We need  
10    wire      inv;  
11  
12    assign inv = sdata ^ CRC[14];  
13  
14    always @(posedge clk) begin  
15        if (idle) begin  
16            CRC = 0;  
17        end  
18        else begin  
19            CRC[14] = CRC[13] ^ inv;  
20            CRC[13] = CRC[12];  
21            CRC[12] = CRC[11];  
22            CRC[11] = CRC[10];  
23            CRC[10] = CRC[9] ^ inv;  
24            CRC[9] = CRC[8];  
25            CRC[8] = CRC[7] ^ inv;  
26            CRC[7] = CRC[6] ^ inv;  
27            CRC[6] = CRC[5];  
28            CRC[5] = CRC[4];  
29            CRC[4] = CRC[3] ^ inv;  
30            CRC[3] = CRC[2] ^ inv;  
31            CRC[2] = CRC[1];  
32            CRC[1] = CRC[0];  
33            CRC[0] = inv;  
34        end  
35    end  
36  
37 endmodule
```

# CRC 15 Area



Stuffing bit 不計入CRC check

傳送的資料 會得出一CRC數值，接續傳送。

# CRC Error

接收的資料連同接收到的CRC數值執行CRC check，  
正確會得到 0 (ZERO)

發出Error (Active/Passive) Frame  
Error Counter +1



# ACK/ACK error



傳送端會送出2個Recessive Bit，

接收端要在ACK Slot回應Dominant告知傳送端。  
此表示傳送端也要讀回BUS狀態。

若沒有得到回應，在ACK Delimiter後面傳送Error Frame。

# ACK 的相關

BUS上有多個Node，若某個Node發送資料且收到ACK，那麼可以保證所有Node都收到此資料嗎？

Node接收到一資料，但ID不吻合，接收Node會在ACK slot回應Dominant嗎？

設置為Listening mode/ Silent mode的Node會送出ACK或Error Frame？

ACK 不是由TX完成。是由接收Node完成

BUS上可能有多個Node，只要其中一個ACK，發送Node就會認定發送成功。

BUS沒有Node回應(BUS只有一個Node)，即Ack error，同時發送Node的TEC累加， $TEC > 128$ 進入Error Passive。注意，如果一直是Ack Error，TEC不会再累加，不会进入Bus Off。

# Bit Monitor / Bit Error

Node發送每一個bit也要檢查是否與發送一致。  
不一致即發生Bit Error。

不包含:

ID仲裁期間

ACK Slot

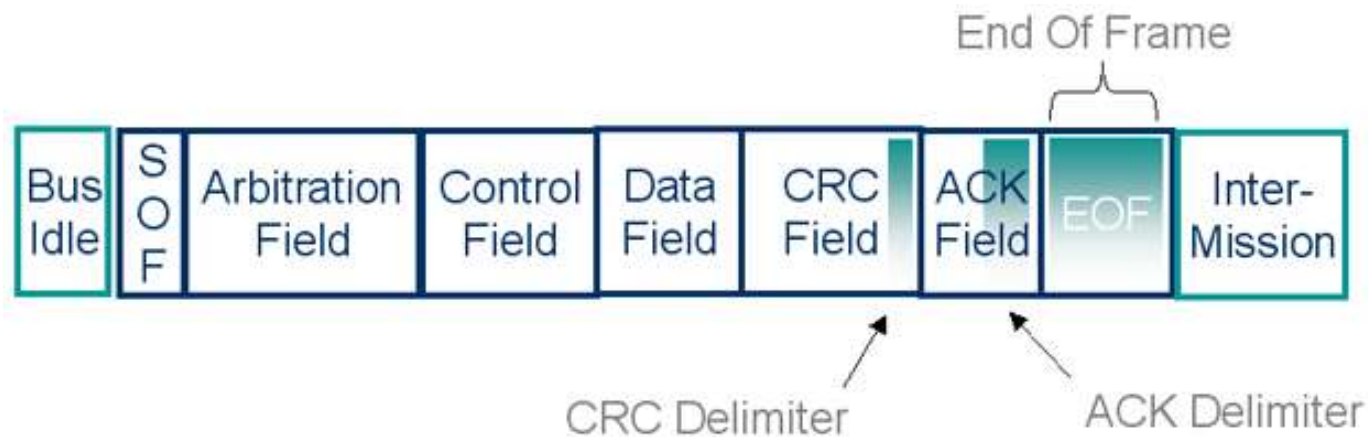
發送Error passive檢測到Dominate。

# Form Error

CRC, ACK Field 都有delimiter，甚至EOF本身就是Data Frame的delimiter，這些bit都預設為recessive(1)，如果在這些bit上讀到dominant(0)，那從下一個bit開始就傳送Error Frame。

不包含:

EOF，Error Frame，Overload Frame 最後一位檢出Dominate。



# Error types (整理)

1. Bit Error ◦
2. Bit Stuffing Error
3. Form Error
4. ACK Error
5. CRC Error

# ID Mask & Filter

其實可以不需要接收每個node的資訊，只要接收跟它有相關的就好了，所以在接收端就需要filter來過濾資料，減輕node負擔

Controller Area Network



# CAN Bus

# 功能定義 需求與設計

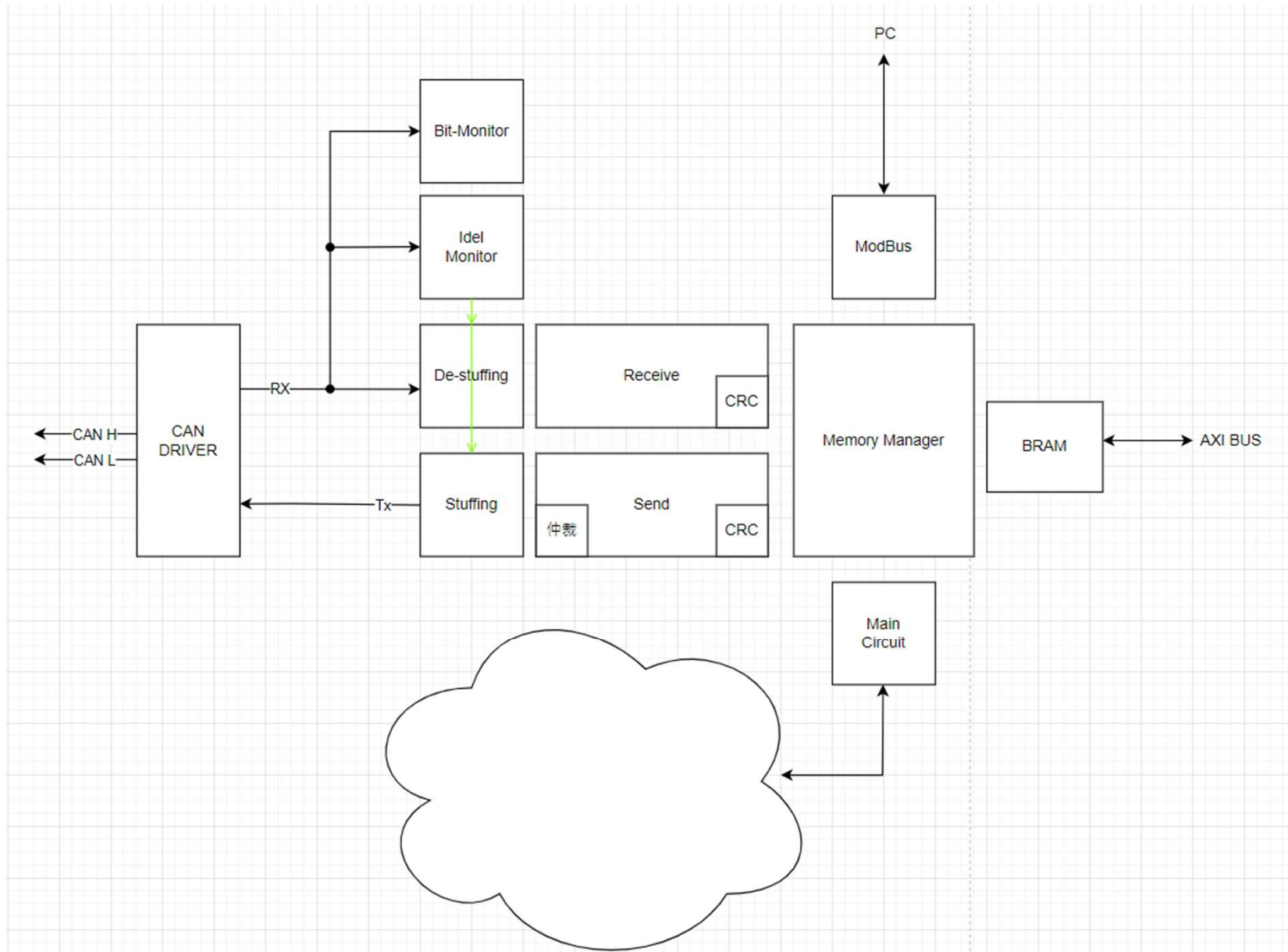
Date Rate: 1Mbps , CAN Classic 。  
System Clock : 100M/50MHz  
1 BRAM Space (36Kb)

與Modbus 相似

1. 寫入一個Byte資料。
2. 寫入一區塊連續資料。
3. 讀取一區塊連續資料。



# Data Flow



# CAN BUS 是否空閒?

BUS空閒時，就可以嘗試發送。

接收到連續11bit( $\geq 11$ )為Recessive :1 表示BUS為IDLE

BUS上狀態是Dominant時，即為非IDEL。

由IDLE變換為非IDLE那一刻，即為SOF位置，也是Clock同步的時機點。



使用CAN 分析儀發送CAN 資料  
FPGA接收資料，利用iLA與電路先分析資料