

Universidade Federal de Pernambuco  
Redes Automotivas IF747

## Relatório

Wallace Soares  
Roger Luiz

Dezembro - 2017

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Diagrama de Blocos</b>	<b>2</b>
<b>3</b>	<b>Maquinas de Estado</b>	<b>3</b>
<b>4</b>	<b>Bit Timing</b>	<b>4</b>
<b>5</b>	<b>Design</b>	<b>5</b>
5.1	Frame Maker . . . . .	5
5.2	Overload and Error Frame Maker . . . . .	5
5.3	Type Frame . . . . .	5
5.4	Stuff . . . . .	5
5.5	ERROR Block . . . . .	5
5.5.1	Bit Stuffing Error Block . . . . .	6
5.5.2	EOF Error Block . . . . .	6
5.5.3	Form Error Block . . . . .	6
<b>6</b>	<b>Test Bench</b>	<b>7</b>
<b>7</b>	<b>Waveforms</b>	<b>7</b>
<b>8</b>	<b>RTLs</b>	<b>7</b>
<b>9</b>	<b>Referências</b>	<b>8</b>

# 1 Introdução

## 2 Diagrama de Blocos

## 3 Maquinas de Estado

## 4 Bit Timing

Cada bit é dividido em 4 partes: SYNC\_SEG, PROP\_SEG, PHASE\_SEG1, PHASE\_SEG2. O ponto onde o valor do bit é lido no barramento é chamado de *sample point*, como indicado na figura abaixo.

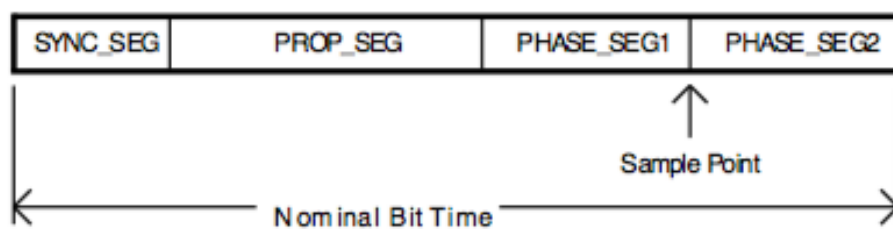


Figura 1: *Bit*

## 5 Design

### 5.1 Frame Maker

### 5.2 Overload and Error Frame Maker

### 5.3 Type Frame

### 5.4 Stuff

### 5.5 ERROR Block

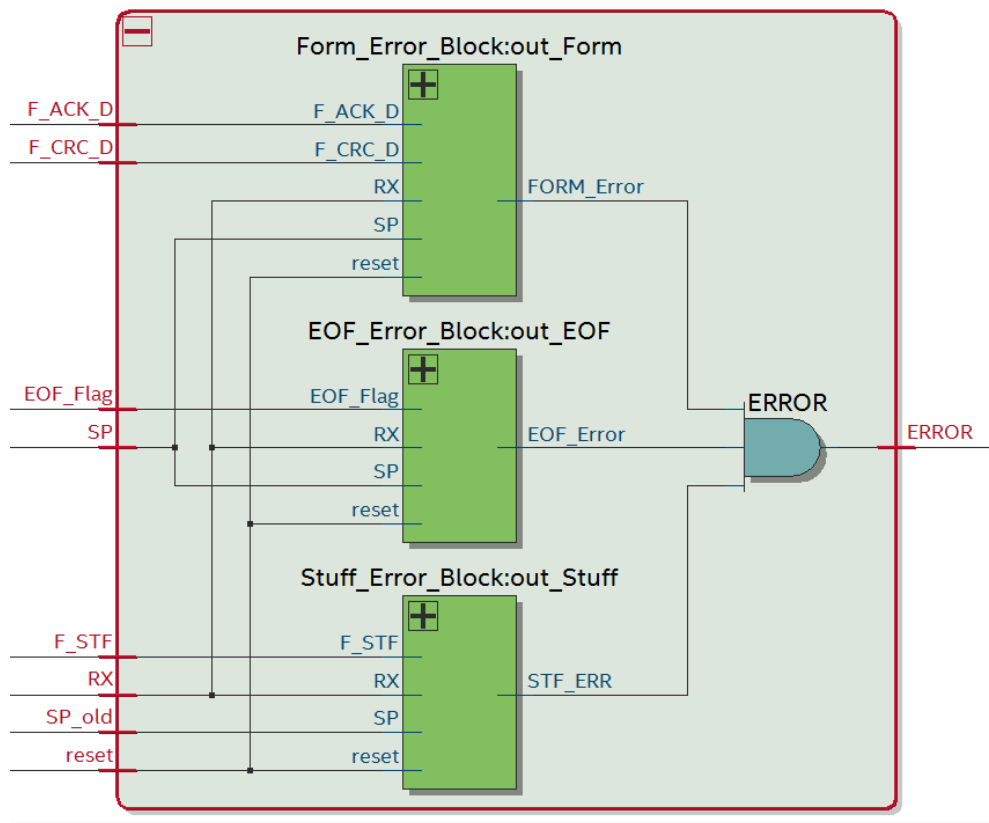


Figura 2: *Error Block*

Este bloco é responsável por detectar os erros passivos que foram determinados na especificação. Com exceção do do erro de CRC que por questões de prazos e projeto decidiu-se por não ser implementado. Este bloco consegue detectar Erros de Bit Stuffing, EOF e erros de forma no CRC e ACK

Delimiter. Cada um desses erros vai ser detalhadamente explicado nas suas respectivas sessões.

A saída dos blocos de erro se liga diretamente ao uma porta AND, e isto se deve ao fato do protocolo CAN considerar 0 lógico como aplicação de DDP entre o CAN High e CAN Low e o valor lógico 1 como a falta de DDP.

Cada um destes blocos recebe uma flag respectiva ao tipo de erro que tem que ser monitorado. Por exemplo, durante os campos em que pode ocorrer stuff a flag F\_STF que permite a detecção de stuff estará com um valor dominante.

### 5.5.1 Bit Stuffing Error Block

Como já explicado anteriormente um stuff ocorre quando cinco bits de mesma magnitude - sendo eles dominantes ou recessivos - ocupam o barramento de forma intermitente. Quando este comportamento é percebido, devido a característica NRZ - Non Return to Zero do protocolo CAN um edge é forçado para que ocorra a sincronização, neste caso Hard Synchronization. Este comportamento é seguido pelo frame entre o SOF - Start of Frame até o CRC Sequence. Caso em algum momento um sexto bit seja detectado nesta região, o bloco BitStuffError deixa como saída um bit dominante que irá informar ao Frame Maker a ocorrência do erro.

### 5.5.2 EOF Error Block

EOF - End of Frame é caracterizado por uma sequência de sete bits recessivos. Este campo indicará que chegamos ao final do frame e assim não ocorrendo nenhum erro até o último bit, toda a operação de transmissão no barramento foi bem sucedida. É dito até o último bit pois o último bit de cada frame é tratado como *don't care*, isto é, caso ocorra um bit dominante no último bit de cada frame um erro de EOF não será detectado.

Um Erro de EOF é detectado quando sua flag, EOF\_Flag, está ativada e um bit dominante é detectado neste campo. Caso ocorra, um bit dominante será liberado na saída deste bloco para que o Frame Maker note o erro e o Frame Maker passe a tratar o erro.

### 5.5.3 Form Error Block

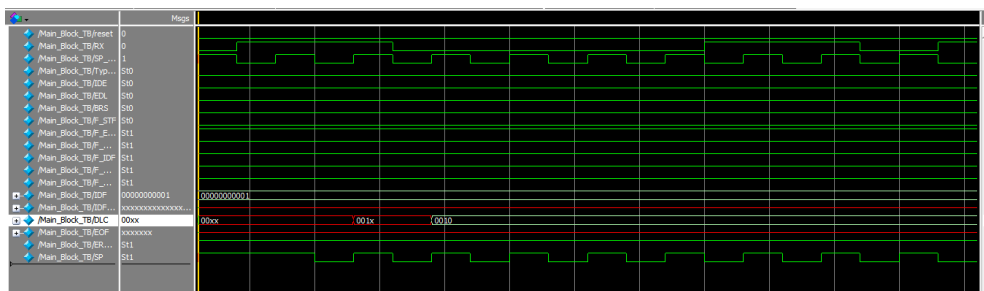
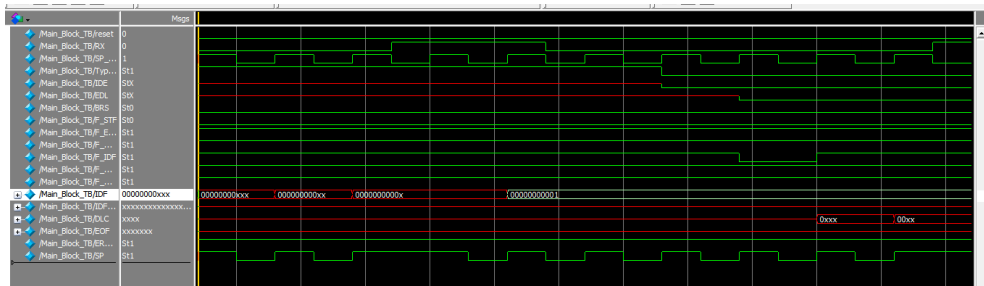
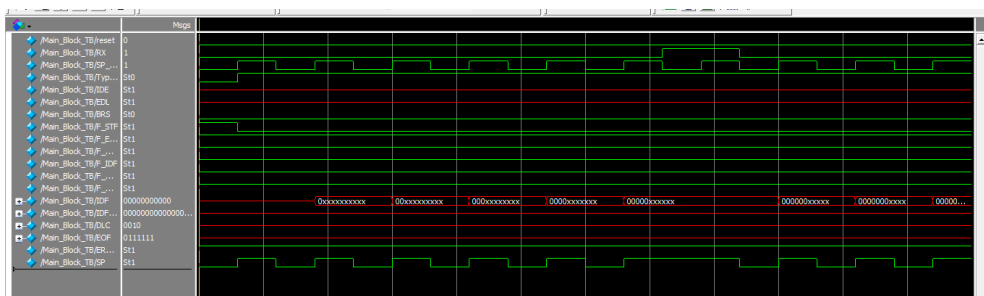
Form Error Block é a unidade do bloco de erro que detecta erros de forma. Apesar do EOF também ser considerado um erro de forma, por decisão de projeto considerando uma melhor separação modular de códigos, os erros de forma no ACK Delimiter e no CRC Delimiter ficaram separados do erro de forma do EOF. No caso para erros de forma no ACK e CRC Delimiter, o



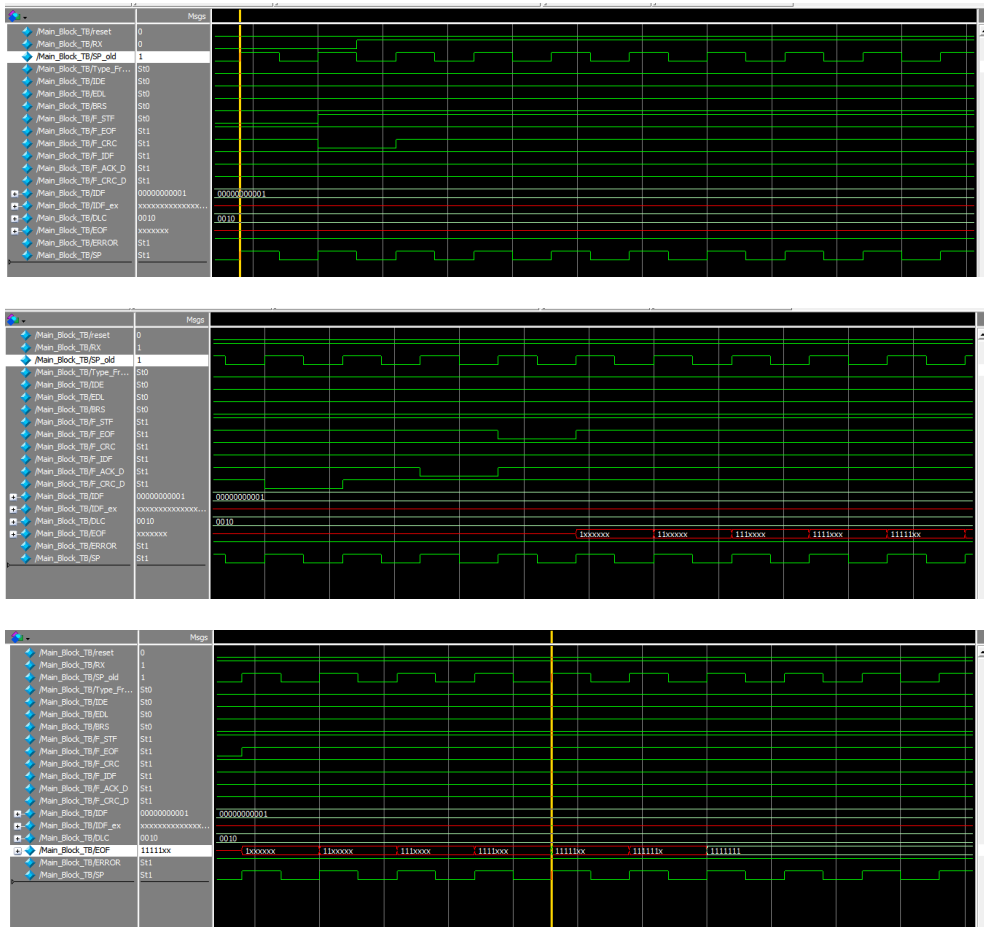
erro é detectado quando a flag esta ativada e um bit dominante é notado, pois ACKs e CRCs Delimiters devem ser obrigatoriamente recessivos.

## 6 Test Bench

## 7 Waveforms



## 8 RTLs



## 9 Referências

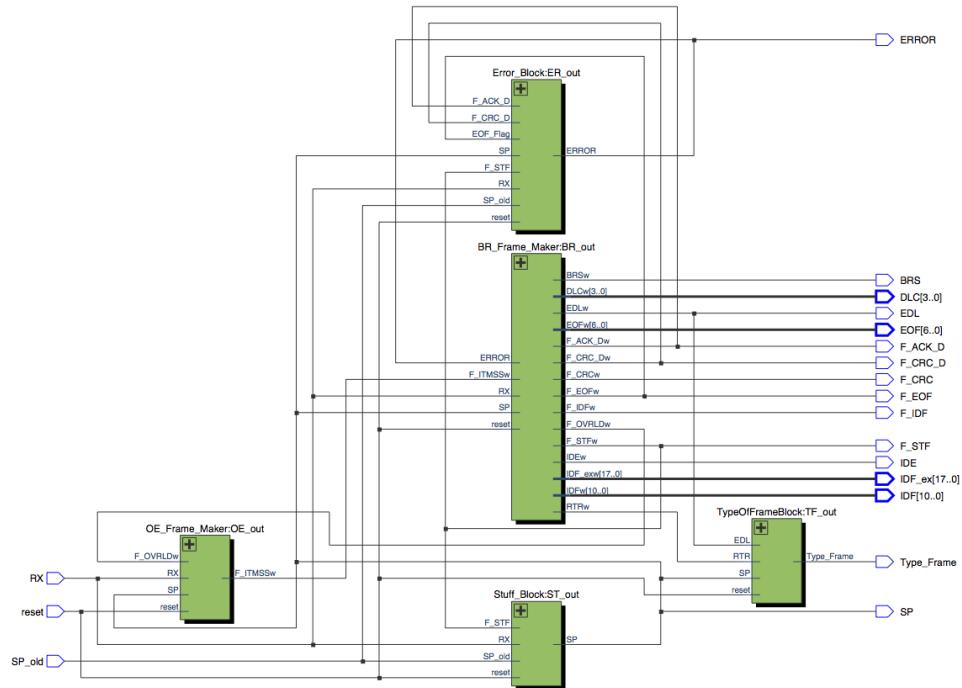


Figura 3: Main Block

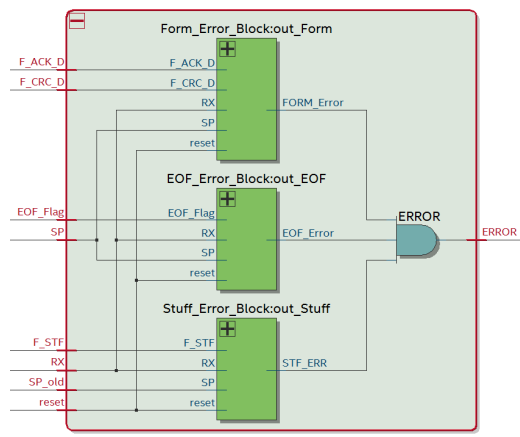


Figura 4: Error Block