TOTEM electronics
Coincidence Chip

Pierre Chalmet

8 March 2006
The goals of the Coincidence Chip

Carry out coincidences over different detector planes to reduce the number of trigger bits.

Needs to be radiation tolerant.

Programmable to deal with different detector geometries.
CC input grouping

Number of Planes (NP > 5)
8 coordinates per plane
yielding 8 outputs

Each group corresponds to one plane

Number of Planes (NP <= 5)
16 coordinates per plane
yielding 16 outputs

Inp 1-8 8 8 8 8
Inp 9-16 8 8 8 8
Inp 17-24 8 8 8 8
Inp 25-32 8 8 8 8
Inp 33-40 8 8 8 8
Inp 41-48 8 8 8 8
Inp 49-56 8 8 8 8
Inp 57-64 8 8 8 8
Inp 65-72 8 8 8 8
Inp 73-80 8 8 8 8
Inp 1-16 16 16 16 16
Inp 17-32 16 16 16 16
Inp 33-48 16 16 16 16
Inp 49-64 16 16 16 16
Inp 65-80 16 16 16 16
CC architecture

I2C interface controls all logic blocks and clock propagation
CC architecture

I2C interface controls all logic blocks and clock propagation
Synchronisation

- Three programmable ways
  - Programmable number of clock cycles
- Definition of the three input ways controlled by the Sync register

<table>
<thead>
<tr>
<th>Sync2</th>
<th>Sync</th>
<th>Resulting function for Sync</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Stretch without Monostable</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Monostable and Stretcher</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Only mask and polarity</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Not used</td>
</tr>
</tbody>
</table>

- One synchronization module per input channel which can be programmed to stretch a pulse over a number of clock cycles and/or to synchronize the pulse with the clock.
- Polarity is defined for all signals at the same time
- Every channel can be masked individually
Synchronisation

CMOS IN 80x

Mask And Polarity

Monostable

Mux.

Stretcher

Mux.

CMOS OUT 80x

I2C interface for logic programming

Three programmable ways

Programmable number of clock cycles

Definition of the three input ways controlled by the Sync register

<table>
<thead>
<tr>
<th>Sync2</th>
<th>Sync</th>
<th>Resulting function for Sync</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Stretch without Monostable</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Monostable and Stretcher</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Only mask and polarity</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Not used</td>
</tr>
</tbody>
</table>

- One synchronization module per input channel which can be programmed to stretch a pulse over a number of clock cycles and/or to synchronize the pulse with the clock.
- Polarity is defined for all signals at the same time
- Every channel can be masked individually
Synchronisation

- One synchronization module per input channel which can be programmed to stretch a pulse over a number of clock cycles and/or to synchronize the pulse with the clock.
- Polarity is defined for all signals at the same time.
- Every channel can be masked individually.

<table>
<thead>
<tr>
<th>Sync2</th>
<th>Sync</th>
<th>Resulting function for Sync</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Stretch without Monostable</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Monostable and Stretcher</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Only mask and polarity</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Not used</td>
</tr>
</tbody>
</table>

Definition of the three input ways controlled by the Sync register.

- Bit of polarity and Mask
- Three programmable ways
- Programmable number of clock cycles

CMOS IN

CMOS OUT

I2C interface for logic programming

Sync2

Sync

Resulting function for Sync

Stretch without Monostable

Monostable and Stretcher

Only mask and polarity

Not used
Synchronisation

One synchronization module per input channel which can be programmed to stretch a pulse over a number of clock cycles and/or to synchronize the pulse with the clock.

Polarity is defined for all signals at the same time.

Every channel can be masked individually.

<table>
<thead>
<tr>
<th>Sync2</th>
<th>Sync</th>
<th>Resulting function for Sync</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Stretch without Monostable</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Monostable and Stretcher</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Only mask and polarity</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Not used</td>
</tr>
</tbody>
</table>

Definition of the three input ways controlled by the Sync register.
CC architecture

LVDS IN

Synchro

V out of NP

Or 1

W out of NP

And/Or

And/Or 2

Or 2

Z out of 16 or 8

Counter

I2C interface for logic programming

Clock propagation

16x

1x

CLK LVDS/CMOS IN

I2C

80x

LVDS OUT

CLK LVDS/CMOS OUT

1x

Or 2

I2C interface controls all logic blocks and clock propagation
Example NP=10

10 detectors planes = 10 inputs for V out of NP

V out of NP

V and NP are specified in the I2C

Yes or Not
solution choice

**Problems:**

- We estimate that the delay time must be less than 2ns.
- The space used must be smallest as possible

**Solutions:**

Three ways have been studied:

- Analog: not sufficiently fast and insufficient margin
- Logic synthesis: the space used is too large
- Full custom logic: adopted solution
**Problem**: the delay time must be inferior than 2 ns.

**Principle**: All the solution are encoded in a word of 4 bits. So we compare this value with the reference which is specified in the I2C. This comparison is done by two parallel adders. So we use the carry of the adder.

In fact we do a subtraction.

\[
0011 \rightarrow \text{reference specified in the I2C and the complement is 1100}
\]

If the value which has been coded is 0010,

\[
\begin{align*}
0010 \\
+ & \quad 1100 \\
= & \quad 01110
\end{align*}
\]
Problem: the delay time must be inferior than 2 ns.

Principle: All the solution are encoded in a word of 4 bits. So we compare this value with the reference which is specified in the I2C. This comparison is done by two parallel adders. So we use the carry of the adder.

In fact we do a subtraction.

0011 -> reference specified in the I2C and the complement is 1100

If the value which has been coded is 0100.

\[
\begin{array}{c}
0100 \\
+ 1100 \\
= 10000
\end{array}
\]
V out of NP

Number of Planes (NP < 5)

I2C interface for logic programming
V out of NP

Number of Planes (NP < 5)

I2C interface for logic programming
V out of NP

- Reference = 2
- Delay time = 1.3 ns
- Consumption = 300 μA (approximate value)
VoutofNP

Number of Planes (NP < 5)

I2C interface for logic programming

MUX.

Référence

3 bits adder

3 bits adder

3

VoutofNP

5x decoder

5x decoder

3

results

NP

VoutofNP
VoutofNP

- NP <= 5 => 16 group of 5
- Reference = 2
- Delay time = 1.5 ns
- Consumption = 600 μA (approximate value)
Number of Planes (NP > 5)

I2C interface for logic programming

MUX.

Result

Référence

NP

Vout of NP
VoutofNP

- NP > 5 => 8 group of 10
- Reference = 2
- Delay time = 2,1 ns
- Consumption = 600 μA (approximate value)
CC architecture

I2C interface controls all logic blocks and clock propagation
Inclusion of neighbors

10 detectors planes = 10 inputs for V out of NP

V out of NP

Yes or Not

No information

V and NP are specified in the I2C

Needed for extra tolerance on alignment and non-perpendicular particles
Inclusion of neighbors

10 detectors planes = 10 inputs for W out of NP

The number of neighbors is specified in the I2C

W and NP are specified in the I2C

Yes or Not
• NP =<5 or >5
• Number of neighbors = 0
• Delay time = 0.8 ns
• Approximate Consumption = 400\mu A for 16 inputs
OR1

- NP =<5
- Number of neighbors = 3
- Delay time = 0.8 ns
- Approximate Consumption = 400μA for 16 inputs
• NP > 5
• Number of neighbors = 3
• Delay time = 0.8 ns
• Approximate Consumption = 400μA for 16 inputs
• Total consumption = 2 mA
CC architecture

I2C interface for logic programming

Clock propagation

<table>
<thead>
<tr>
<th>AO1</th>
<th>AO2</th>
<th>Resulting function for And/or</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>(V out of NP) and (W out of NP) (default)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>(V out of NP) or (W out of NP)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>V out of NP</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>W out of NP</td>
</tr>
</tbody>
</table>
CC architecture

I2C interface controls all logic blocks and clock propagation

<table>
<thead>
<tr>
<th>LO1</th>
<th>LO2</th>
<th>Resulting function for And/or2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Output and not (Z out of 8 or 16) (default)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Output and (Z out of 8 or 16)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Not(Output and not (Z out of 8 or 16))</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Not(Output and (Z out of 8 or 16))</td>
</tr>
</tbody>
</table>
CC architecture

LVDS IN 80x Synchro → V out of NP → And/Or → And/Or 2 → Or 2 → LVDS OUT 16x

CLK LVDS/CMOS IN I2C interface controls all logic blocks and clock propagation → CLK LVDS/CMOS OUT

I2C Interface for logic programming

Clock propagation

<table>
<thead>
<tr>
<th>O2_2</th>
<th>O2_1</th>
<th>O2_0</th>
<th>Resulting function for Or2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Output all outputs (default)</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Group outputs in groups of 2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>Group outputs in groups of 4</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Group outputs in groups of 8</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Group outputs in groups of 16</td>
</tr>
</tbody>
</table>
CC architecture

Every outputs can be linked to a counter to count the number of pulse.

The counter size is programmable and is specified in the I2C.

<table>
<thead>
<tr>
<th>CT(1)</th>
<th>CT(0)</th>
<th>Counter Reset</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>256</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>65536</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>16777216</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>4294967296</td>
</tr>
<tr>
<td>Control Register 0</td>
<td>Control Register 1</td>
<td>Control Register 2</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>7 6 5 4 3 2 1 0</td>
<td>7 6 5 4 3 2 1 0</td>
<td>7 6 5 4 3 2 1 0</td>
</tr>
<tr>
<td>- - CT CT - Sync Sync NP</td>
<td>OV OV OV NP V V V V</td>
<td>Z Z Z W W W W</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Organization of the I2C register
Other Characteristics

- Layout Space = 24.9 mm²
- Delay time without synchronisation = 12 ns
- Consumption = 400 mA (The main consumption comes from the pads ring)
- We can control the evaluation moment (rising edge or falling edge of CLK)
- Diode protection ×2 (outputs)
Coincidence chip

- Pin 1
- Inp 17-32
- Inp 1-16
- I2C
- Outp 9-16
- Outp 1-8
- Inp 33-48
- Inp 49-64
- Inp 65-80
- Bias & Clk

5022.5 µm

4960.00 µm

CHALMET Pierre
Coincidence chip

- Inp 17-32
- Inp 1-16
- I2C
- Outp 9-16
- Outp 1-8
- Inp 33-48
- Inp 49-64
- Inp 65-80
- Bias & Clk

Pitch = 100 µm
Pitch = 50 µm
LVDS REPEATER

LVDS INPUTS → LVDS/CMOS INPUT PAD → CMOS SIGNAL → BUFFER → CMOS SIGNAL → CMOS/LVDS OUTPUT PAD → LVDS OUTPUTS

16x

LVDS INPUTS

16x
LVDS REPEATER

- Total consumption: 112 mA
- Delay time: 1.2 ns
- Area = 6.24 mm²
- Diode protection ×4