

Simulação do Sistema de Leitura ECAL

Isabel M C Teixeira

Novembro de 2000

LIP
Lisboa

Roteiro

Objectivo do trabalho

Documentos analisados

Objectivos da simulação

Funcionalidade do sistema a simular (como foi entendida)

Aspectos da funcionalidade não compreendidos correctamente

Validação da funcionalidade a simular

Próximos passos

Objectivo do trabalho

Identificar a funcionalidade do DCC Readout e o seu enquadramento no sistema ECAL

Identificar as restrições de temporização e a possibilidade de ocorrência de situações de overflow

Modelar o DCC Readout em termos de blocos constituintes, interligações, funcionalidade e desempenho

Utilizar, para o efeito, a ferramenta Rose RT (Rational)

Documentos base

“Descripton of the Data Concentrator Card (DCC) for the CMS-ECAL”, José Carlos Da Silva e João Varela

“ECAL Front-End and Trigger Primitive Generators System High-Level Modelling”, Amélia Santos, Nuno Vitorino, Pedro Moreno, José Casquinha, Isabel Teixeira, João Varela

Documento descritivo dos formatos de dados de:

DCC Block
Rose Block
Pointer Block
Tower Block
Crystal Block

(documento distribuído na última reunião no LIP- Julho 2000)

Objectivos da simulação - genéricos

Validar o funcionamento do sistema em termos de funcionalidade e de desempenho, em condições de funcionamento correcto e observar o funcionamento face à ocorrência de erros.

Para tal, é necessário:

modelar a arquitectura do sistema em termos de:

blocos constituintes

interligações

máquinas de estado associadas aos diferentes blocos

operações executadas em cada estado

lista de sensibilidades

Objectivos da simulação - específicos

Simular o comportamento do sistema, próximo de, ou no, limite das restrições de temporização

para tal é necessário:

**avaliar os limites das restrições de temporização
condições em que ocorrem**

Isto permitirá determinar:

Como não perder a identidade dos dados

**Como decidir o compromisso : velocidade de
comunicação/ dimensões dos buses no event builder**

Condições de overflow

Índice de ocupação das RAMs

**Pontos (do circuit) e condições (temporais) de
estrangulamento**

Funcionalidade do sistema

Objectivo funcional do ECAL Data Concentrator - DCC

em cada L1A trigger:

Colecta os dados provenientes do conjunto de placas ROSE do seu crate

Valida os dados que chegam

Envia os dados válidos (eventos) para os canais :

ECAL DAQ
Trigger DAQ

Actualiza os diferentes registos de erros (se necessário)

Distribuição da funcionalidade do sistema

Em alto-nível:

2 placas - Interface (DCC-IF) e mother board (DCC-MB)

Interface (em cada L1A):

Recebe dados dos links de dados provenientes de Upper-level readout e Trigger boards

Armazena os dados em memórias

Mother- board:

Controlador e interface de dados dos diferentes blocos do DCC

Funcionamento do DCC Readout

Os dados provenientes dos módulos de readout chegam por links (ponto a ponto) e são armazenados nas FIFOs de entrada - iFIFO

os dados transferidos para as iFIFO são controlados pelos input_handlers

o event-builder verifica a integridade dos dados provenientes de cada módulo de leitura, processa-os e envia:

- event-data para o oFIFO ECAL-DAQ

 - (cópia para um oFIFO separado - para testes)

- trigger data para o oFIFO TRIGGER-DAQ

Funções do Event Builder

Verifica a integridade de todos os 17 segmentos de dados que chegam das placas ROSE, ao ritmo L1A para
garantir a sincronização L1A
monitorizar erros (e, (em caso de overflow ???), gerar eventos vazios
Monitoriza a ocupação das iFIFOs e das oFIFOs

Serve de Buffer para os dados a enviar para os links: DAQ e TRIGGER

Espera a chegada dos dados de cada ROSE, correspondentes a um dado L1A

Verifica a integridade do Header:

- Identidade do Header
- Identidade (#) da carta
- Word count value
- Card event ID
- Headcheck sum

Funções da iFIFO Handler

Recebe um start command (via TTC ou VME)

monitoriza a Data Present Flag enviada por cada ROSE (antes dos dados ou quando????)

se estiver activa, os dados são válidos e devem (MUST) ser escritos nas iFIFO

Actualiza o # de ciclos escritos (Handler-WC-Value) (quando termina a escrita de dados)

O Handler-WC-Value vai ser comparado, (posteriormente) com o Word-Count-value no Header dos dados provenientes da ROSE

Actualiza o registo L1A_waiting e fixa o endereço de entrada da iFIFO para o próximo L1A

(dados prontos para ser processados pelo output-handler do event builder)

Funções da oFIFO Handler

Quando chega um trigger L1A, espera que os módulos de readout escrevam os dados nas iFIFO

Verifica a integridade dos dados de cada canal de dados, em cada trigger L1A. Verifica:

- 1- Integridade do Header
- 2- Card-Number-ID
- 3 - Word Count Value
- 4 - Card-Event-ID
- 5 - Header-check-sum

SE não detectar erros:

constroi os eventos de saída para ECAL-DAQ e TRIGGER-DAQ
decrementa o registo L1A-Waiting. Se este não estiver vazio, recomeça a geração de novo evento de saída

Funções da oFIFO Handler (cont.)

Se detectar erros em 1 ou 2:

- actualiza o registo de erros correspondente

- escreve 'empty event' na oFIFO

- ignora o resto dos dados provenientes deste canal

- move-se para o próximo endereço da oFIFO e repete as operações

Se detectar erros em 3-5, actualiza os registos de erro correspondentes.

Monitoriza a ocorrência de overflow nas iFIFO

Ligação ao 'Trigger Data Concentrator'

Toda a informação relacionada com o Trigger está localizada na mesma secção do DAQ

Em cada L1A (par???) os dados do TPG presentes no DAQ event path são copiados para um buffer (pequeno) que agrupa os 17 segmentos de dados num evento que é propagado por um link para um Trigger data concentrator

Informação sobre erros, overflow e FM control são propagados da mesma forma

DCC ADDRESS Section

Espaço utilizado - A32/D32 do VME (DMA)

2 sub-espacos de endereçamento:

Sub Address Region (SAR) - descodificação de endereços dos blocos principais

Select Circuit Register (SCR) para acesso directo à região SAR seleccionada

Comunicação:

Bus PCI-CPU

Ligação directa ao CPU próximo do DCC

VME data bus - usado para os valores dos registos (ignorado para comandos directos)

Proximos passos

Descrição do sistema no Rose RT
modelo lógico
estrutural
protocolos

**Simulação do sistema para diferentes tipos de dados,
diferentes cenários e diferentes velocidades**

**Deformação de cenários para determinar os limites das
restrições temporais impostas por caminhos criticos**