**2.3.1. Módulo Contador de Eventos**

O módulo contador é implementado como um circuito programável através da firmware da unidade FPGA. Ele é responsável pela contagem do número de pulsos digitais vindos da eletrônica de Front-End, da saída do discriminador. Cada um desses pulsos corresponde a um sinal analógico na saída de uma SiPM que é processado por um dos doze canais da FEE.

Os cintiladores são organizados em três grupos de quatro tiras cintilantes. Os canais de 1 a 4 formam o grupo A, os canais de 5 a 8 formam o grupo B, e os canais de 9 a 12 formam o grupo C. A firmware realiza a contagem de pulsos individuais em cada um dos canais e também realiza contagens de todas as possíveis coincidências duplas, triplas e quádrupla entre os canais de um mesmos grupo.

O FPGA deriva seu sinal de clock de um cristal oscilador de 50 MHz, representado pelo nome LOCALCLK. Na firmware, é utilizado um módulo PLL, que a partir do LOCALCLK deriva como output um sinal de 100 kHz, usado como clock do bloco de controle dos módulos DAC do circuito, e um sinal repetido de 50 MHz, que tem o papel de sinal de clock de todos os demais blocos lógicos.

Devido a instabilidades no sinal da fotomultiplicadora e ruídos de chaveamento no discriminador é possível que um único pulso vindo do detector gere a detecção de diversas bordas de subida no pino de entrada correspondente da FPGA. Para evitar que isso aconteça, cada um dos doze canais digitais de entrada é alimentado em um bloco denominado Gate Timer. Quando ocorre uma borda de subida em um desses canais, a saída correspondente nesse bloco é mantida em alto por um tempo configurável chamado nT\_GATE. A duração do gate tem o valor de um número inteiro de ciclos de clock que vai de 1 a 255.

Os doze sinais de saída do Gate Timer então chegam no bloco de coincidências. Esse bloco é responsável por determinar as coincidências de detecção em canais de um mesmo grupo. Isso é implementado alimentando cada sinal de um grupo em portas AND organizadas de acordo com cada uma das combinações possíveis de dois, três ou quatro canais. Assim, por exemplo, para o grupo A, temos os canais A1, A2, A3 e A4. Com isso, as combinações possíveis são:

* Combinações duplas: A1A2, A1A3, A1A4, A2A3, A2A4 e A3A4;
* Combinações triplas: A1A2A3, A1A2A4, A1A3A4 e A2A3A4;
* Combinação quádrupla: A1A2A3A4;

Os grupos B e C apresentam combinações análogas entre seus canais. Assim temos 6 combinações duplas, 4 combinações triplas e uma quádrupla por grupo. A imagem abaixa detalha o esquema de lógica combinacional:

IMAGEM DO COINCIDENCE BLOCK

Finalmente, os pulsos individuais e os pulsos de coincidência são direcionados para a entrada do bloco de contagem. Com 11 combinações mais 4 pulsos individuais, são 15 contagens por grupo, num total de 45 contagens. Cada contagem é armazenada em 3 bytes de memória, com todas as contagens ocupando em todo 135 bytes.

Quando é detectada uma borda de subida em uma das 45 entradas, a contagem correspondente é incrementa em uma unidade. As contagens vão sendo atualizadas pela duração do ciclo de contagem atual. Uma entrada de RELEASE é responsável por sinalizar o fim do ciclo em curso. Quando RELEASE sofre uma borda de descida, cada valor de cada contagem é descarregado em saídas correspondentes do bloco, para depois serem armazenadas no bloco de memória da firmware. As contagens então voltam para zero, e inicia-se um novo ciclo.

O bloco de memória é formado pelos bancos A e B. Ele é acessível através de um bloco de comunicação serial via protocolo I2C para o usuário que, no caso deste experimento, é o microcontrolador ESP32. O banco A é acessível apenas para leitura, sendo responsável por armazenar as 45 contagens de pulsos vindas do bloco de contagem. Já o banco B é acessível para leitura e escrita, e armazena variáveis de configuração e de status, que são usadas para mediar e sincronizar a comunicação entre o usuário e a FPGA.

Cada banco possui 255 bytes de memória, podendo ser endereçados por um único byte. Assim os endereços de memória vão de 0x00 a 0xFF. Os cinco endereços mais altos, de 0xFF a 0xFB, são comuns aos dois bancos, representando, nessa ordem, a versão da firmware, o reset de software, a seleção de banco, e os endereços de FLAG e ACK. O banco A é completamente reservado para contagens de pulsos. O banco B armazena variáveis para configuração do período de um ciclo de contagem de pulsos, o TAQ (*time of acquisition*), o número de ciclos de clock em um período do Gate Timer, o nT\_GATE, o reset do bloco de controle dos módulos DAC, e o controle dos LEDs do módulo FPGA para testes.

Os endereços de FLAG e ACK servem para sincronizar a comunicação entre o FPGA e o microcontrolador. Quando um ciclo de aquisição, ou de contagem, chega ao fim, o valor do FLAG se torna 0x01. O ESP32, que monitora continuamente o FLAG, interpreta isso como sinal de que as últimas contagens estão prontas para serem lidas. O ACK é um endereço reservado para escrita pelo microcontrolador, e é setado para 0x01 quando todas as contagens já foram lidas. Com isso, a firmware seta o FLAG e, em seguida, o ACK, ambos de volta para 0x00.

A arbitragem da duração do ciclo de aquisição, da atualização de variáveis de configuração, e da comunicação com o microcontrolador é feita pelo bloco de controle. Ao começo de um novo ciclo, ele lê da memória o valor mais recente configurado para o TAQ e o nT\_GATE, sendo esse enviado para o Gate Timer. Então, um bloco interno de cronometragem marca a passagem do número de segundos desde o início do ciclo. Quando esse número se iguala ao valor do TAQ, o sinal de saída de RELEASE vai para alto, fazendo o bloco de contagem descarregar as contagens no banco A da memória. Por fim, o endereço de FLAG é setado, o RELEASE volta para zero, e um novo ciclo se inicia.

DESCREVER O BLOCO DE CONTROLES DOS DACS