

Correção de PCR integrada à adaptação de taxa para processadores de fluxos de transporte MPEG-2

Heitor J. Savino¹, Eddie B. L. Filho^{1,2}

¹Universidade Federal do Amazonas – UFAM
Manaus, Amazonas, Brazil, 69077-000

²Centro de Ciência, Tecnologia e Inovação do Pólo Industrial de Manaus – CT-PIM
Manaus, Amazonas, Brazil, 69057-040

heitor_savino@ufam.edu.br, eddie@ctpim.org.br

Abstract. *The MPEG-2 transport stream is used in many systems for transporting multiplexed audio, video and data. The Program Clock Reference (PCR) is a stamp used to provide time bases for the transmitted programs and is also responsible for synchronizing the receiver. Due to the processing chain between transmitter and receiver, which often modifies the relative distance between time stamps, the PCR data may suffer from inaccuracies referred to as jitter. This paper introduces a method for joint rate adaptation and PCR correction, which also has the potential to reduce the output jitter caused by the null packets insertion, in the rate adaptation module. Simulation results show that the proposed method is effective and keeps the jitter level, at the output, close to the one at the input.*

Resumo. *O fluxo de transporte MPEG-2 é usado, em vários sistemas atuais, para a transmissão de áudio, vídeo e dados. A Referência de Relógio de Programa (Program Clock Reference - PCR) é uma informação utilizada para fornecer bases de tempo aos programas transmitidos e é também responsável por sincronizar o receptor. Devido à cadeia de processamento entre transmissor e receptor, que muitas vezes modifica a distância relativa entre as referências de tempo, os valores de PCR geralmente apresentam imprecisões conhecidas como jitter. Este artigo introduz um método que executa a correção de PCR e a adaptação de taxa conjuntamente, com o objetivo de reduzir o jitter de saída causado pela inserção de pacotes nulos, no módulo de adaptação de taxa. Resultados de simulações mostram que o método proposto é eficaz e mantém o nível de jitter, na saída, próximo ao da entrada.*

1. Introdução

A Referência de Relógio de Programa (*Program Clock Reference - PCR*) é a informação de tempo utilizada pelo Fluxo de Transporte (*Transport Stream - TS*) MPEG-2 para a sincronização do receptor, de modo que o áudio e o vídeo sejam corretamente apresentados. Em resumo, esta informação é um retrato do estado atual do contador de 27 MHz do transmissor e fornece toda a informação de tempo necessária ao sistema. Quando este valor é igual ao do contador do receptor, o sistema está perfeitamente sincronizado; caso contrário, uma malha amarrada por fase (*Phase Locked Loop*

- PLL) é usada na adaptação do relógio local, através da diferença dos valores de PCR [ISO/IEC 13818-1 2007].

Entretanto, a cadeia de processamento entre transmissor e receptor pode causar desvios na informação de PCR, o que é conhecido como *jitter* [Tryfonas and Varma 1999] e pode comprometer o processo de sincronização. Por exemplo, durante a remultiplexação, demultiplexação ou conversão de um tipo de interface em outra [ETSI EN 50083-9 2003], as posições relativas dos valores de PCR podem mudar, causando *jitter*. De modo a manter o erro dentro de limites aceitáveis, um módulo de correção de PCR é normalmente empregado, com o objetivo de compensar o desvio devido às novas localizações, no TS, dos valores de PCR. O padrão MPEG-2 estabelece um limite máximo de ± 500 ns [ISO/IEC 13818-1 2007] para o *jitter*, o que assegura o correto funcionamento do receptor.

Alguns trabalhos na literatura [Lee et al. 1997, Xingdong et al. 2002, Longfei et al. 2002, He et al. 2007, He et al. 2005, Chen et al. 2009] fornecem soluções para a correção de PCR, modificando os seus valores de modo a compensar as mudanças ocorridas no módulo de adaptação de taxa. Dois esquemas básicos podem ser identificados: um conjunto de contadores dedicados [Lee et al. 1997, He et al. 2007], onde cada contador é associado a um programa no TS e é diretamente carregado com o respectivo valor de PCR, e um único contador [Longfei et al. 2002, Xingdong et al. 2002], do qual referências intermediárias são tomadas para se realizar a compensação. Ambos apresentam vantagens e desvantagens, e a melhor escolha depende da aplicação-alvo.

Até o momento, as soluções propostas apenas modificam os valores de PCR para que os mesmos reflitam suas novas posições no TS, porém, nenhuma delas aborda o descasamento entre a adaptação de taxa e a granularidade da correção de PCR, o que é dado principalmente pela relação entre a taxa de saída, a taxa de entrada e a frequência do relógio de 27 MHz. Devido a isso, mesmo quando a correção do PCR é realizada corretamente, ainda há um acréscimo no *jitter*.

Com base nisso, é possível supor que, se o algoritmo de adaptação de taxa operar de forma cooperativa com o módulo de correção de PCR, o sistema será capaz de reduzir o *jitter* inserido no TS adaptado. Esta é a inspiração para o presente trabalho, que propõe uma nova estrutura com correção de PCR integrada à adaptação de taxa, de modo que pacotes com PCR sejam transmitidos apenas quando o *jitter* previsto for suficientemente pequeno.

Este artigo está organizado da seguinte forma. A Seção 2 descreve os métodos tradicionais, destacando suas principais características. Na Seção 3, a nova estrutura para a correção de PCR integrada à adaptação de taxa é apresentada. Finalmente, a Seção 4 fornece os resultados das simulações e a Seção 5 as conclusões do trabalho.

2. Métodos tradicionais de correção de PCR

Se algum dispositivo ou rede altera a cadência dos dados entre codificador e decodificador, tal variação tem o potencial de causar *jitter*, o que pode levar a desvios na frequência do relógio de 27 MHz do decodificador. De modo a evitar essa situação, os valores de PCR devem ser corrigidos, o que normalmente é feito através da adição de um termo de

compensação, como em

$$PCR_{corr} = PCR + \Delta PCR_{comp}, \quad (1)$$

onde PCR_{corr} é o novo valor de PCR, PCR é o valor original retirado do TS e ΔPCR_{comp} é o termo de compensação. Esta é a ideia básica utilizada nos métodos tradicionais, que são apresentados a seguir.

2.1. O método dos contadores dedicados

Um TS MPEG-2 pode conter vários programas, cada um com a sua própria base de tempo, o que caracteriza um Fluxo de Transporte com Múltiplos Programas (*Multiple Program Transport Stream - MPTS*) [ISO/IEC 13818-1 2007]. Com isso, uma solução intuitiva para o problema de correção de PCR consistiria na utilização de um número de contadores de 27 MHz igual ao número de programas no TS [Lee et al. 1997]. Entretanto, alguns programas podem apresentar a mesma base de tempo, sendo que o importante é o número de bases de tempo diferentes.

O primeiro passo nesta abordagem consiste em identificar e extrair a informação de tempo dos pacotes contendo PCR. Este dado é então carregado em um contador associado ao programa do qual o pacote faz parte, no próximo gatilho do relógio de 27 MHz, incrementado por 1 ($PCR + 1$). Assim que a adaptação de taxa é feita e o pacote é agendado para envio, o conteúdo do contador referente ao pacote é recuperado e usado para substituir o valor de PCR original. A arquitetura deste sistema é mostrada na Figura 1.

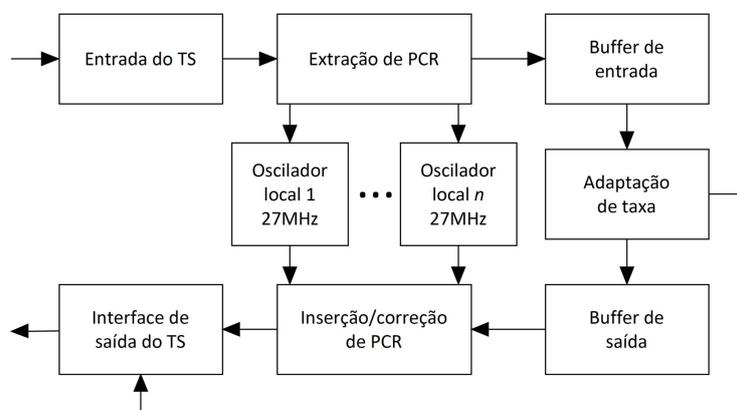


Figura 1. Diagrama de bloco do método dos contadores dedicados.

2.2. O método da compensação

Durante o processo de correção de PCR realizado no método da compensação [Longfei et al. 2002, Xingdong et al. 2002], apenas um contador de 27 MHz é utilizado, o que economiza um grande número de recursos. Quando um pacote contendo PCR é detectado, o estado atual do contador ($Count_{in}$) é tomado e subtraído do respectivo valor de PCR, sendo que o resultado desta operação (PCR_I) então substitui o PCR original no pacote. Quando o respectivo pacote é enviado pelo sistema, o PCR armazenado é mais uma vez alterado: o estado atual do contador ($Count_{out}$) é novamente tomado e adicionado ao valor do PCR. Desta forma, cada valor de PCR original é substituído por um

novo valor de referência, calculado como:

$$PCR_I(i) = PCR(i) - Count_in(i) \quad (2)$$

$$PCR_{corr}(i) = PCR_I(i) + Count_out(i) \quad (3)$$

$$PCR_{corr}(i) = PCR(i) + \Delta PCR_{comp}(i), \quad (4)$$

onde $\Delta PCR_{comp}(i) = Count_out(i) - Count_in(i)$ é o termo de compensação para o valor $PCR(i)$.

O método da compensação pode ser visto como uma correção em duas etapas, na qual um contador de 27 MHz mede o tempo de processamento de pacotes no módulo de adaptação de taxa e compensa o atraso variável sofrido pelos mesmos. Este método é simples e utiliza uma quantidade de recursos mínima, porém, apresenta a desvantagem de realizar várias adições de 42 bits, o que aumenta a carga computacional do sistema. O diagrama de blocos deste esquema de correção pode ser visto na Figura 2.

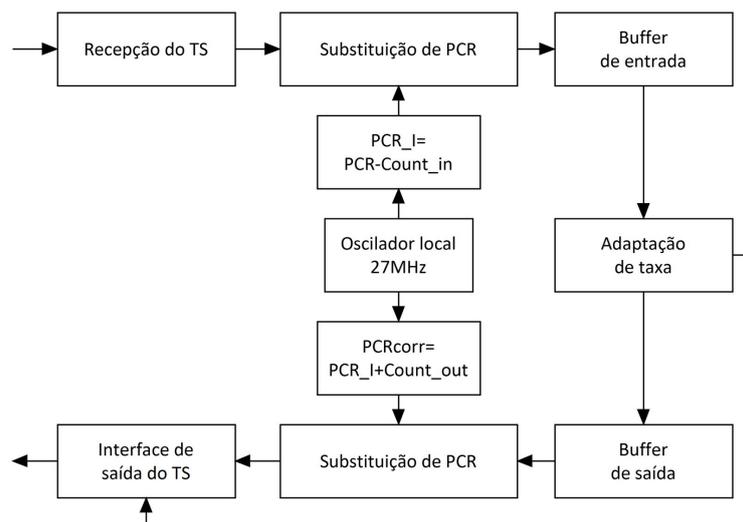


Figura 2. Diagrama de blocos do método da compensação.

3. A nova estrutura para a correção de PCR integrada à adaptação de taxa

Os métodos tradicionais apresentados na seção anterior, quando corretamente implementados em um processador de TS, como um remultiplexador ou um demultiplexador, são capazes de corrigir os valores de PCR, evitando uma quebra no processo de sincronização. Entretanto, comparando-se o TS de saída com o de entrada, ainda há um acréscimo inerente de *jitter*, na saída, o que é dado principalmente pelo descasamento entre a taxa de saída, a taxa de entrada e a frequência do relógio de 27 MHz.

Por exemplo, é possível se adaptar a taxa de um TS, com pacotes contendo 188 bytes, de 12 Mbps de entrada para 48 Mbps de saída. Como a correção de PCR é dada por um relógio local de 27 MHz, um pacote demoraria um número inteiro de períodos de *clock* no adaptador de taxa, isto é, $((8 \cdot 27000000)/48000000) \cdot 188 = 846$. Dado que a distância entre dois valores de PCR em um TS é sempre um número inteiro de pacotes

[ISO/IEC 13818-1 2007], o processo de correção de PCR compreenderia um número inteiro na contagem do relógio de 27 MHz. Logo, se o *jitter* na entrada for muito pequeno ou 0, dependendo da implementação, é bastante provável que este permaneça baixo.

Num outro cenário, se a conversão é de 12 Mbps para 20 Mbps, o número de períodos de *clock* por pacote fica $((8 \cdot 27000000)/20000000) \cdot 188 = 2030,4$. Por exemplo, quando 27 pacotes são enviados entre dois pacotes com PCR, tal ação pode resultar em um erro durante a correção do PCR, de aproximadamente 30 ns. Além disso, se também houver erro na próxima correção de PCR, isso poderá causar uma diferença de dois ou mesmo três períodos de *clock* de 27 MHz, no intervalo entre dois valores de PCR. Desta forma, se vários dispositivos com tal comportamento processarem um dado TS, o *jitter* resultante poderá prejudicar o processo de sincronização. No entanto, se 35 pacotes forem enviados entre esses dois pacotes, o erro resultante tenderá a permanecer próximo de 0, dado que $((8 * 27000000)/20000000) \cdot 188 \cdot 35 = 71064$. Mesmo que o relógio do transmissor não seja exatamente 27 MHz, a diferença é normalmente bastante pequena, podendo-se assumir um *jitter*, na saída, bastante reduzido.

Uma outra observação interessante, ainda relativa à correção de PCR, consiste no fato de que o momento no qual o valor de PCR é carregado, para correção, não está sincronizado com o relógio de 27 MHz. Isto pode resultar em mais uma fonte de *jitter*, com o potencial de comprometer ainda mais a correção.

A análise apresentada mostra que, se o módulo de adaptação de taxa estiver ciente do mecanismo de correção de PCR e operar de forma conjunta com o mesmo, o *jitter* no TS de saída poderá ser mantido próximo ao encontrado no TS de entrada.

3.1. O atraso da entrada com relação ao relógio de 27 MHz

Dado um TS com taxa de bits constante, o intervalo de tempo entre dois pacotes consecutivos, em períodos de *clock* de 27 MHz, é dado por $((8 \cdot 27000000)/Taxa\ de\ entrada) \cdot 188$. Este valor pode indicar um número não inteiro (como mostrado na seção anterior), significando que, cada vez que um pacote contendo PCR for carregado no *buffer* de entrada, o início da contagem do relógio local (relacionado ao termo ΔPCR_{comp} da Equação 1) não estará necessariamente sincronizado com a chegada do pacote, causando erro na correção do valor de PCR.

Conhecendo-se a taxa de entrada do TS, é possível estimar a defasagem entre a chegada do pacote e o gatilho do contador. Por exemplo, para um taxa de entrada de 20 Mbps, o tempo entre pacotes consecutivos é dado por $((8 \cdot 27000000)/20000000) \cdot 188 = 2030,4$ períodos de 27 MHz. Sendo assim, cada vez que um pacote é carregado no *buffer* de entrada, a diferença entre o gatilho de *clock* e o instante da chegada do pacote é aumentada em 0,4 períodos.

O adaptador de taxa, então, opera contando o número de pacotes que chegam entre PCRs, através de um contador de pacotes relacionado à taxa de entrada. Cada vez que um pacote com PCR é carregado no *buffer*, o adaptador de taxa toma o valor desse contador para calcular o erro inserido; em seguida, o contador é reiniciado. O atraso é dado por este valor de contador, multiplicado pelo valor do tempo de chegada de pacotes, em períodos de 27 MHz, e tomando-se apenas a parte fracionária (*e.g.* para 326 pacotes, o atraso será de $2030,4 \cdot 326 = 661910,4$, ou seja, 0,4 períodos).

3.2. O atraso da saída com relação ao relógio de 27 MHz

A interface de saída do processador de TS também contribui para o *jitter*, através da diferença entre o gatilho do contador de 27 MHz e o instante em que o pacote é enviado. Com uma taxa de dados constante, o atraso por pacote pode ser calculado pela parte não inteira de $((8 \cdot 27000000)/Taxa\ de\ saída) \cdot 188$. Assim, para uma taxa de saída de 25 Mbps, cada pacote na saída causará um erro de 0,32 períodos, dado que um pacote na saída demora $((8 \cdot 27000000)/25000000) \cdot 188 = 1624,32$ períodos de *clock* de 27 MHz.

Vale ressaltar que o tempo que um pacote passa no adaptador de taxa, desde o instante em que o mesmo é carregado no *buffer* de entrada até o momento de ser enviado pela interface de saída (representado por ΔPCR_{comp} da Equação 1), deve ser representado por um número inteiro de períodos de *clock* de 27 MHz. Com isso, é possível se inserir o menor *jitter* possível, conforme mostrado no início da seção.

Do mesmo modo que se calcula a diferença entre o gatilho do contador de 27 MHz e o instante de chegada de um pacote no *buffer* de entrada, um contador para os pacotes de saída também é utilizado. Desta forma, contam-se pacotes enviados entre PCRs, sendo que o contador é zerado toda vez que um pacote contendo PCR é enviado. Assim, é possível estimar a diferença inserida entre o gatilho do contador de 27 MHz e o instante em que um pacote é enviado, tomando-se o valor do contador e multiplicando-se o mesmo pelo número de períodos de *clock* de um pacote na saída (e.g. para 408 pacotes, o atraso será de $1624,32 \cdot 408 = 662722,56$, ou seja, 0,56 períodos).

3.3. A correção de PCR integrada à adaptação de taxa

No início desta seção, foi mostrado que, se o envio de pacotes resultar em um número não inteiro de períodos de *clock*, tal fato pode resultar em uma fonte adicional de *jitter*. Como a correção de PCR está relacionada à quantidade de gatilhos do contador de 27 MHz, que um pacote passa no adaptador de taxa, é interessante fazer com que esse intervalo de tempo seja o mais próximo possível de um valor inteiro. A ideia fundamental da correção de PCR integrada à adaptação de taxa, que é proposta neste trabalho, consiste em estimar o *jitter* que seria introduzido quando pacotes com PCR são enviados, de modo a agendar a sua transmissão para o momento em que o *jitter* for mínimo, ou seja, o tempo no qual o número de períodos do *clock* de 27 MHz for inteiro.

Quando um pacote contendo PCR chega ao *buffer* de entrada, toma-se o valor do contador de pacotes de entrada (que indica a quantidade de pacotes que chegaram desde o último PCR, conforme a Subseção 3.1), de modo a se calcular a diferença de fase entre o gatilho do contador de 27 MHz e a chegada de pacotes na interface de entrada. Assim, quando for necessário enviar o mesmo pacote, o módulo de adaptação de taxa tomará o valor do contador de saída (pacotes que foram enviados desde o último PCR) para agendar o seu envio, fazendo com que o erro inserido entre o gatilho do contador de 27 MHz e o instante da saída do pacote seja o mais próximo possível do que foi medido na entrada. Conforme o cenário apresentado na Subseção 3.1, o atraso inserido na entrada seria de 0,4 períodos. O adaptador de taxa, então, enviaria o pacote conforme a Subseção 3.2, compensando com um atraso de saída igual a 0,56 períodos. Desse modo, poder-se-ia dizer que o pacote foi enviado com um erro de contagem de 0,16 períodos de *clock*.

Após transmitir o pacote contendo PCR, o adaptador de taxa reinicia o contador de pacotes enviados e guarda a diferença *diff_out_in* entre os atrasos inseridos na saída

e na entrada, que no exemplo anterior seria de 0,16. Assim, quando o próximo pacote com PCR for enviado, o módulo de adaptação de taxa tentará igualar o atraso de saída a de entrada, subtraído pelo valor de $diff_out_in$, isto é,

$$desired_out = estimated_in - diff_out_in, \quad (5)$$

onde $desired_out$ é o valor de atraso desejado na saída e $estimated_in$ é o atraso identificado pela contagem dos pacotes de entrada.

É interessante notar que $diff_out_in$ representa a diferença de tempo entre os instantes de chegada e de envio dos pacotes, com relação ao clock de 27 MHz. Sendo assim, esse valor é iniciado com zero, dado que o sistema começa a enviar pacotes assim que o primeiro é detectado no *buffer* de entrada.

O adaptador de taxa envia o pacote armazenado sempre que o *buffer* de entrada indica dois ou mais pacotes armazenados; caso contrário, o mesmo envia pacotes nulos. Quando um pacote com PCR é agendado para ser enviado, o módulo adaptador de taxa calcula quantos pacotes nulos devem ser enviados antes do mesmo, de modo que o atraso inserido na saída seja o mais próximo possível de $desired_out$. Assim, gera-se um vetor *holding_vector*, que indica o *jitter* de saída para n pacotes, onde n varia de 0 (pacote enviado imediatamente) até um número máximo dado por

$$max_hold = \frac{(B_{size} - Pack_n)}{PCR_n} \cdot \frac{Taxa_de_saida}{Taxa_de_entrada}, \quad (6)$$

onde B_{size} , $Pack_n$, e PCR_n são o tamanho do *buffer* do módulo de adaptação de taxa, o número de pacotes já armazenados e o número de pacotes, no *buffer*, contendo PCR, respectivamente. Logo, max_hold é o número de pacotes que podem ser enviados sem que o *buffer* estoure. Além disso, max_hold não pode ser maior que um número predefinido, de modo a garantir que tais cálculos sejam executados em tempo hábil.

A Tabela 1 apresenta um exemplo de adaptação de taxa, em uma conversão de 20 Mbps para 25 Mbps, mostrando contadores, atraso de entrada, valor desejado de saída, valor real conseguido na saída e a diferença $diff_out_in$.

Tabela 1. Exemplo de adaptação de taxa

<i>Contador na entrada</i>	<i>Atraso na entrada</i>	<i>diff_out_int</i>	<i>desired_out</i>	<i>Contador na saída</i>	<i>Atraso na saída</i>	<i>Novo diff_out_int</i>
0	0	0	0	0	0	0
312	0.8	0	0.8	403	0.96	0.16
465	0	0.16	0.84	584	0.88	0.04
466	0.4	0.04	0.36	580	0.6	0.24
466	0.4	0.24	0.16	585	0.2	0.04
468	0.2	0.04	0.16	582	0.24	0.08

3.4. Compartilhando o contador: a estrutura de semáforo

Quando o sistema processa um MPTS, a ideia geral é que se utilize um número de contadores igual ao de programas encontradas no TS, como mencionado na Seção 2.1, ou

então que se empregue o método da compensação apresentado na Seção 2.2. Entretanto, se uma estrutura de acesso ao contador for utilizada, é possível o uso de apenas um contador no sistema, com um número reduzido de adições de 42 bits. O diagrama em blocos da abordagem proposta é mostrado na Figura 3.

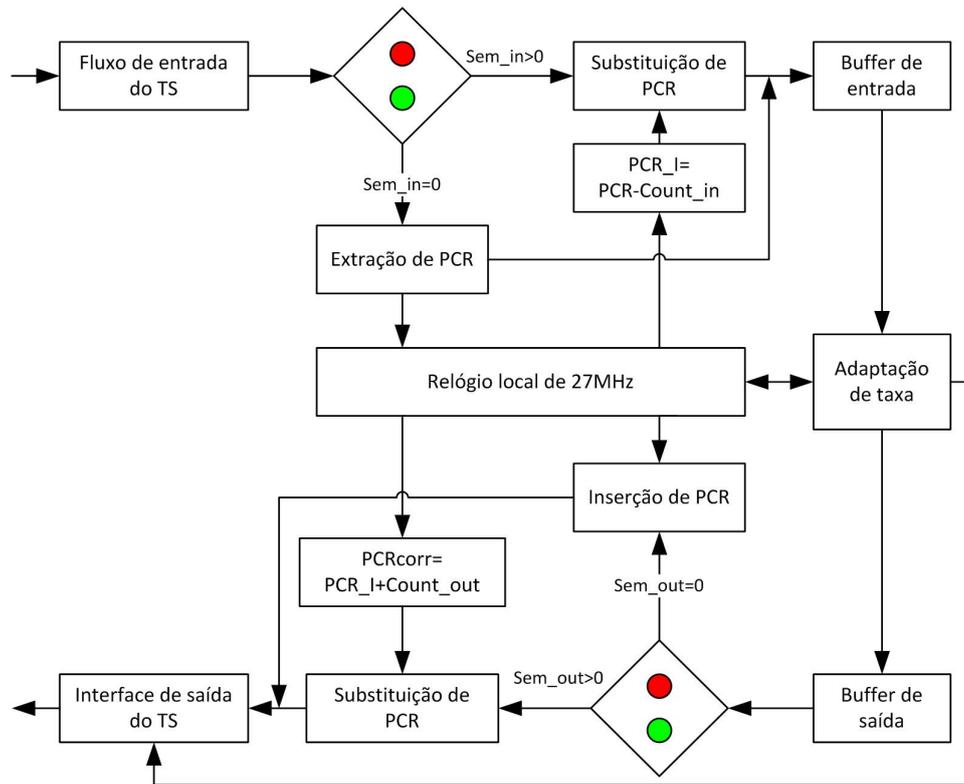


Figura 3. Diagrama de blocos da estrutura de contador controlado por semáforo.

Para se compartilhar o contador entre todos os programas do TS, uma estrutura de semáforo foi criada, com o objetivo de controlar o acesso ao mesmo. Essa lógica auxiliar simples apresenta dois estados: o estado livre, que ocorre quando o acumulador está disponível para ser carregado de forma direta, com valores de PCR extraídos de pacotes que chegam pela interface, e o estado ocupado, indicando que o contador já foi carregado com algum valor de PCR e que as próximas correções serão feitas através do método da compensação. Neste último caso, ao invés de serem usados contadores dedicados para a compensação dos valores de PCR, o sistema gerará uma referência temporária para este pacote, baseada no valor de PCR do mesmo e do contador, de acordo com a Equação 1. Quando este for selecionado para envio, pelo adaptador de taxa, o processo de correção será finalizado aplicando-se a Equação 2. No entanto, se o contador estava no estado livre quando este pacote chegou ao *buffer* de entrada, o valor do contador é apenas copiado diretamente para o campo de adaptação de taxa do pacote.

Conforme ilustrado na Figura 3, dois semáforos são utilizados: um para a entrada e outro para a saída, ambos iniciados com 0. Sempre que um pacote com informação de PCR chega ao *buffer* de entrada, o valor do semáforo de entrada é incrementado; da mesma forma, quando um pacote com PCR deixa o módulo de adaptação de taxa, o valor do semáforo de saída é incrementado. Assim, quando o semáforo de entrada apresenta o valor 0, o contador pode ser diretamente carregado; caso contrário, referências

temporárias são tomadas para a compensação. Além disso, quando o semáforo de saída está em 0, o valor do contador pode ser copiado diretamente no pacote; caso contrário, a compensação é aplicada. Se os dois semáforos apresentam valores iguais, o que significa que todos os pacotes com PCR já foram enviados, ambos são reiniciados com 0, indicando, mais uma vez, que o contador se encontra no estado livre.

O módulo completo de correção de PCR também pode ser visto como duas máquinas dependentes, sendo uma para a entrada e para a saída do sistema. O pseudo-algoritmo para a entrada é:

1. Verificar Sem_in ;
2. Se $Sem_in > 0$, ir para 4;
3. Carregar o valor de $PCR(i)$ no contador e ir para 5;
4. Fazer $PCR(i) = PCR(i) - Count_in(i)$;
5. Incrementar Sem_in ;
6. Fim.

e o pseudo-algoritmo para a saída é:

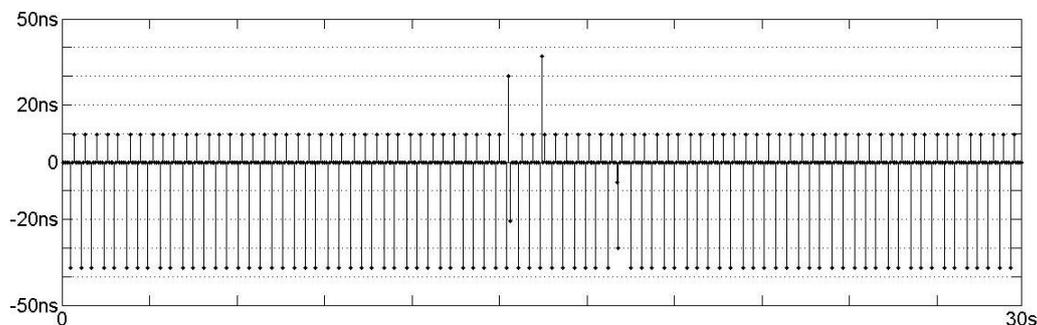
1. Verificar Sem_out ;
2. Se $Sem_out > 0$, ir para 4;
3. Ler o valor de $PCR(i)$ do contador e ir para 5;
4. Fazer $PCR(i) = PCR(i) + Count_out(i)$;
5. Incrementar Sem_out ;
6. Se $Sem_out \neq Sem_in$, ir para 8;
7. Fazer Sem_out e Sem_in iguais a 0;
8. Fim.

4. Resultados de simulações

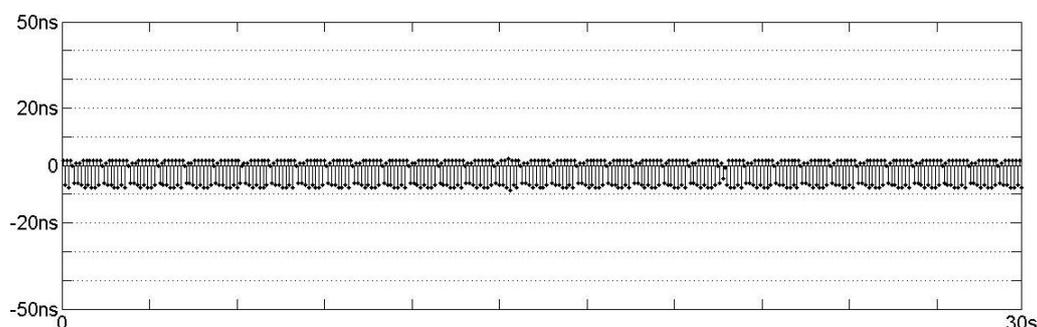
Para fins de análise do método, foram realizadas simulações no ambiente Real Time Workshop/Simulink, com dois cenários. No primeiro caso, 30 segundos de um TS no padrão brasileiro, contendo 2 programas com *jitter* zero, foram processados por dois conversores SPI para SPI [ETSI EN 50083-9 2003]. O primeiro conversor usou a adaptação de taxa integrada à correção de PCR e o segundo o método da compensação [Longfei et al. 2002]. A taxa de dados do TS é de 32.507936 Mbps, com 15.52 Mbps para o primeiro programa e 600 Kbps para o segundo (o restante é composto pelas tabelas de SI/PSI [Reimers 2004] e por pacotes nulos). As medições de PCR foram feitas segundo a norma DVB para acurácia de PCR [ETSI TR 101 290 2001]. Para se realizar uma comparação com outros métodos, a literatura geralmente apresenta apenas o gráfico de *jitter* e o valor máximo inserido. Os resultados referentes ao primeiro programa são mostrados na Figura 4, para se avaliar o *jitter* de saída do sistema. A taxa de saída é fixada em 43 Mbps, com o intuito de se obter um número não inteiro de períodos de clocks, por pacote (conforme mostrado na Subseção 3.2), a qual é alcançada através da inserção de pacotes nulos.

Nos dois conversores, o módulo de adaptação de taxa é composto por um *buffer* de entrada, um *buffer* de saída e um algoritmo de escalonamento. Entretanto, no primeiro, os pacotes que chegam são armazenados em um *buffer* de entrada e, toda vez que um pacote com PCR deve ser enviado para o *buffer* de saída, a adaptação de taxa

integrada à correção de PCR busca o melhor momento para executar esta tarefa. No segundo, por sua vez, um pacote é enviado sempre que há mais de 2 pacotes no *buffer* de entrada. A Figura 4 mostra os resultados obtidos com o método proposto e o método da compensação.



(a) *Jitter* inserido no Programa 1 com adaptação de taxa tradicional.



(b) *Jitter* inserido no Programa 1 com o método proposto.

Figura 4. Resultados de simulação para o Programa 1 do TS escolhido.

Nas Figuras 4(a) e Figura 4(b), é possível observar que o *jitter* de saída, para o método da compensação, é maior que o apresentado pelo método proposto. Além disso, a estrutura de semáforo não chegou a alcançar valores maiores que 1, ou seja, em todo o tempo de simulação, o contador foi utilizado diretamente, sem qualquer operação aritmética. Por outro lado, o método da compensação exigiu 1368 adições (dado que foram processados 684 pacotes) no módulo de correção de PCR (ver Seção 2.2). O *jitter* máximo inserido pelo método proposto, que foi de 8ns, é menor que o apresentado pela abordagem tradicional por compensação [Longfei et al. 2002, Xingdong et al. 2002], que foi de 37ns, o que possibilitaria futuros processamentos no mesmo TS, sem que o processo de sincronização fosse comprometido.

No segundo cenário, 30 segundos de um TS com 14 programas e 8 bases de tempo diferentes, a uma taxa aproximada de 24.1 Mbps, foram processados, analisando-se os resultados referentes aos três primeiros programas. A escolha desse MPTS foi feita com o intuito provocar uma alta carga computacional no adaptador/corretor. As taxas de dados para os programas 1, 2 e 3 são 2.13 Mbps, 2.44 Mbps e 3.11 Mbps, respectivamente. A taxa de saída é fixada em 43 Mbps. Os resultados, para o segundo cenário, são mostrados na Figura 5. Vale ressaltar que, praticamente, não houve aumento no nível de *jitter*, o que faz com que todo o processo de conversão de taxa seja quase transparente.

Durante a simulação, a evolução dos parâmetros do semáforo foi acompanhada

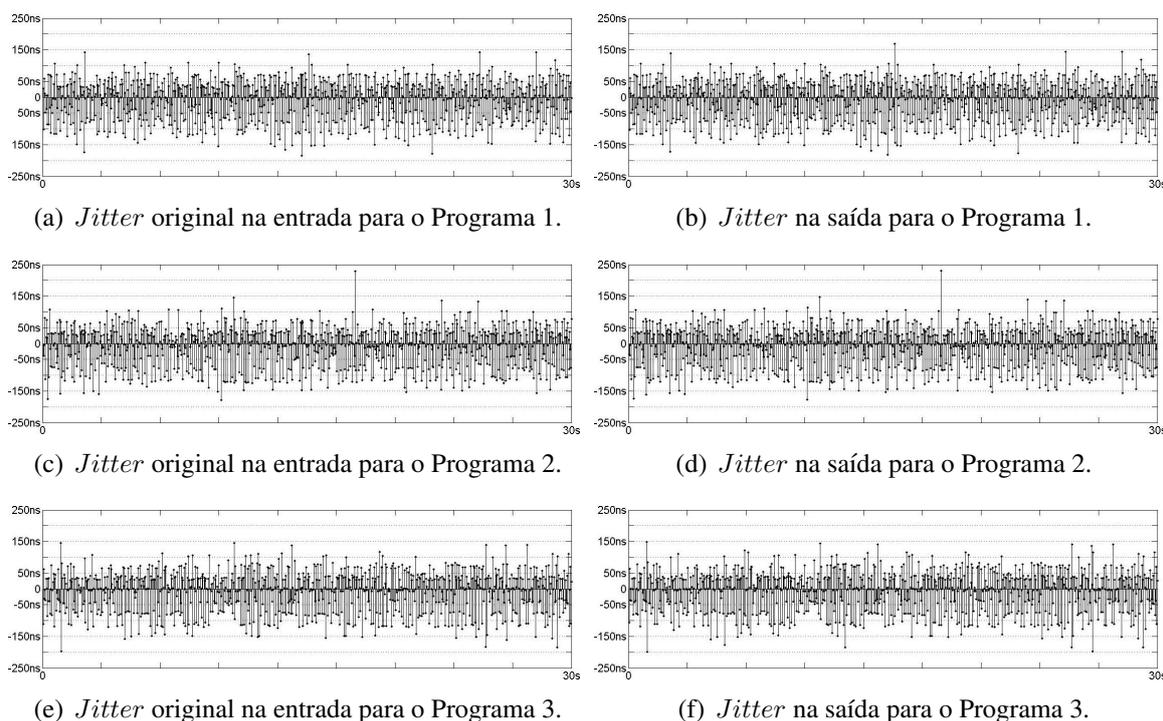


Figura 5. Resultados de simulação para os três primeiros programas do MPTS escolhido.

e analisada. Como resultado, o valor máximo alcançado pelo semáforo foi 5, indicando que, no pior caso, apenas 4 compensadores foram usados. Além disso, 90,3% dos valores de PCR foram carregados diretamente no contador, sendo que os outros fizeram uso de um processo similar ao utilizado pelo método da compensação. Sendo assim, durante a maior parte do tempo, o contador esteve livre para ser carregado com um novo valor de PCR, o que também seria verdade para a maioria dos fluxos de transporte comerciais.

Devido ao método proposto, que é baseado em uma estrutura de semáforos, o módulo de correção de PCR usou de apenas 1536 adições de 42 bits, em vez das 15838 necessárias no método da compensação. Além disso, apenas um contador de 27 MHz foi utilizado, ao invés dos 14 contadores exigidos no método dos contadores dedicados. Dessa forma, a complexidade computacional da correção de PCR pode ser reduzida com o uso de semáforos, em comparação ao método da compensação, e as exigências de hardware são aliviadas, com relação ao método dos contadores dedicados. Entretanto, a lógica adicional necessária para o cálculo do envio de pacotes e para a implementação da estrutura de semáforos, por sua vez, apresenta uma carga adicional de recursos.

5. Conclusão

Um método baseado em correção de PCR integrada à adaptação de taxa foi apresentado, com o intuito de corrigir os valores de PCR e reduzir o *jitter* inserido pelo módulo de adaptação de taxa, que é devido ao reposicionamento dos pacotes durante um processo de conversão de taxa. O novo método introduz menos *jitter*, quando comparado aos esquemas tradicionais, nos quais a adaptação de taxa e a correção de PCR trabalham de forma independente. Isto é obtido através da troca de informações entre os módulos

envolvidos, fazendo-se com que o tempo envolvido no envio de pacotes, entre valores de PCR consecutivos, seja o mais próximo possível de um número inteiro de períodos do relógio 27 MHz. Os resultados mostram que o método é eficaz e pode ser empregado em sistemas que exigem bases de tempo precisas.

Referências

- ISO/IEC 13818-1: Information technology - Generic coding of moving pictures and associated audio information - Part 1: Systems. 3rd ed. Geneva, Switzerland: 2007.
- C. Tryfonas, and A. Varma, "Timestamping Schemes for MPEG-2 Systems Layer and Their Effect on Receiver Clock Recovery", *IEEE Transactions on Multimedia*, Vol. 1, No. 3, pp. 251-263, September 1999.
- ETSI EN 50083-9: Cable distribution systems for television, sound signals and interactive multimedia signals; Part 9: Interfaces for CATV/SMATV Headends and Similar Professional Equipment for DVB/MPEG-2 Transport Streams, 2003.
- S. I. Lee, S. B. Cho, J. H. Kim, H. H. Jeon, and D. G. Oh, "Implementation of MPEG-2 TS Remultiplexer and Data Transport Unit for HDTV Satellite Broadcasting", *IEEE Transactions on Consumer Electronics*, Vol. 43, No. 3, pp. 324-329, August 1997.
- W. Xingdong, Y. Songyu, and L. Longfei, "Implementation of MPEG-2 transport stream remultiplexer for DTV broadcasting", *IEEE Transactions on Consumer Electronics*, Vol. 48, No. 2, pp. 329-334, May 2002.
- L. Longfei, Y. Songyu, and W. Xingdong, "Implementation of a New MPEG-2 Transport Stream Processor for Digital Television Broadcasting", *IEEE Transactions on Broadcasting*, Vol. 48, No. 4, pp. 348-352, December 2002.
- Y. He, J. Zhou, and Y. Zhou, "Counter-Set Based PCR Jitter Correction Method for DVB-T System", in *proc. of the International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 2940-2943, Shanghai, September 2007.
- U. Reimers, "DVB - The Family of International Standards for Digital Video Broadcasting" 2nd ed. Berlin, Germany: Springer, 2004.
- Y. He, J. Zhou, and Y. Zhou, "Implementation of TS De-multiplexer with FPGA in DVB-IP Gateway for Network TV", presented at the *2005 IEEE 7th Workshop on Multimedia Signal Processing*, Shanghai, China, October 2005.
- Y.-P. Chen, T.-N. Chien, P.-H. Cheng, and S.-J. Chen, "An Agile and Low Cost FPGA Implementation of MPEG-2 TS Remultiplexer for CATV Head-end Equipment", in *proc. of the 2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks*, Kaohsiung, Taiwan, December 2009.
- ETSI TR 101 290 V1.2.1 (2001-05): Digital Video Broadcasting (DVB); Measurement guidelines for DVB systems, 2001.