

# Um Contador Controlado por Semáforos Para a Correção de PCR

Heitor J. Savino, Eddie B. L. Filho

**Resumo**—O fluxo de transporte MPEG-2 é muito utilizado na transmissão de áudio, vídeo e dados multiplexados, fornecendo bases de tempo para os programas enviados. Esta informação, conhecida como PCR e também responsável pela sincronização do receptor, pode sofrer erros durante a sua transmissão, devido à cadeia de processamento entre transmissor e receptor. Este artigo introduz um novo método de correção de PCR, baseado em um contador controlado por semáforos, que é capaz de fornecer uma referência de tempo estável para o decodificador. A arquitetura do módulo de correção, que possui características como complexidade computacional e requisitos de hardware reduzidos, é apresentada e comparada a métodos tradicionais. Resultados de simulação mostram que o método proposto é eficaz e mantém um baixo nível de *jitter* na saída do sistema.

**Palavras-Chave**—Correção de PCR, MPEG-2, fluxo de transporte, semáforo

**Abstract**—The MPEG-2 transport stream is used in many systems for transporting multiplexed audio and video in a transparent way, providing time bases for the transmitted programs. Such information, which is called PCR and is also responsible for synchronizing the receiver, may suffer from inaccuracies, due to the processing chain between transmitter and receiver. This paper introduces a new PCR correction method named as the semaphore-controlled counter, which provides a stable time reference for the decoder. The correction module architecture is presented and compared to traditional methods, exhibiting features like low computational complexity and hardware requirements. Simulation results show that the proposed method is effective and provides small *jitter* levels at the system output.

**Keywords**—PCR correction, MPEG-2, transport stream, semaphore

## I. INTRODUÇÃO

O padrão MPEG-2 sistemas [1], que descreve a formação do Fluxo de Transporte (*Transport Stream* - TS), é amplamente utilizado em muitos sistemas de transmissão espalhados pelo mundo (e.g. DVB, ISDB e ATSC). Ele foi concebido para transportar áudio, vídeo e dados através de canais ruidosos e de forma transparente, combinando até mesmo programas com bases de tempo independentes. Quando recebido, o TS proporciona sincronização por meio de referências de tempo, conhecidas como Referência de Relógio de Programa (*Program Clock Reference* - PCR), que controlam o relógio de 27MHz do receptor.

O PCR é um retrato do estado atual do contador de 27 MHz do transmissor. Em um caso ideal, o PCR é igual ao valor presente no contador do receptor; caso contrário, uma malha amarrada por fase (*Phase-Locked Loop* - PLL) adapta o relógio do sistema, através da diferença entre o PCR e o relógio local.

Heitor J. Savino\*, Eddie B. L. Filho\*†. \*Universidade Federal do Amazonas - UFAM, Av. Gen. Rodrigo Octávio Jordão Ramos, 3000, Manaus - AM, 69077-000, Brasil. †Centro de Ciência, Tecnologia e Inovação do Pólo Industrial de Manaus, Rua Salvador, 391, Adrianópolis, Manaus-AM, 69057-040, Brasil. E-mails: heitor\_savino@ufam.edu.br, eddie@ctpim.org.br.

Quando TSs são remultiplexados ou demultiplexados, ou, por exemplo, convertidos de uma interface serial assíncrona (*Asynchronous Serial Interface* - ASI) para uma interface paralela síncrona (*Synchronous Parallel Interface* - SPI) [5], a posição relativa entre os valores de PCR muda, o que pode gerar instabilidade no relógio de sistema; tal perturbação é conhecida como *jitter* [6]. Para se lidar com esse tipo de situação, utilizam-se adaptadores de taxa, que normalmente operam com base na inserção de pacotes nulos e possuem um módulo de correção de PCR.

O controle de *jitter* é de grande importância para decodificadores de precisão (e.g. Set-top Boxes), pois estes usam o relógio de 27MHz, adaptado pelo PCR, para gerar a frequência da subportadora de cor e taxa de amostragem de vídeo [8]. O padrão MPEG-2 estabelece um nível máximo de *jitter* de  $\pm 500$ ns [1], para assegurar o bom funcionamento do receptor.

Alguns trabalhos presentes na literatura [2]–[4], [7], [9], [10] indicam soluções para o problema do *jitter*, que são baseadas na correção dos valores de PCR, durante o processo de adaptação de taxa. Dois esquemas básicos podem ser claramente identificados: um conjunto de contadores dedicados [2], [9], onde cada programa no TS é associado a um contador que carrega diretamente o respectivo valor de PCR, e um contador único [3], [7], a partir do qual são tomadas referências de tempo intermediárias. Ambos apresentam vantagens e desvantagens e, dependendo do cenário (e.g. *Field-Programmable Gate Arrays* - FPGA, etc.), um deles pode ser o mais adequado.

Este trabalho também aborda essa questão, através da proposição de um novo método para a correção de PCR, que apresenta as principais vantagens presentes nos esquemas tradicionais: um contador único, que agora é controlado por uma estrutura de semáforos, e a possibilidade dos valores de PCR serem corrigidos diretamente pelo contador, o que evita adições de 42 bits. Como consequência, este novo método reduz o número de operações durante o processo de correção de PCR e mantém baixo nível de recursos necessários.

Este artigo está organizado da seguinte forma. A seção II apresenta os esquemas tradicionais, destacando suas principais características. Na seção III, o método proposto é descrito, e a seção IV mostra uma breve comparação com outros métodos presentes na literatura. Finalmente, a seção V apresenta os resultados de simulação e as conclusões são expostas na seção VI.

## II. MÉTODOS TRADICIONAIS DE CORREÇÃO DE PCR

Quando algum dispositivo ou rede altera o tempo de entrega da informação de PCR, entre codificador e decodificador, as variações resultantes tendem a causar instabilidades (*jitter*). A fim de corrigir os valores de PCR, normalmente adiciona-se

um termo de compensação, tal como:

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

onde  $PCR_{corr}$  é o novo valor de PCR,  $PCR$  é o valor original e  $\Delta PCR_{comp}$  é o termo de compensação. Esta ideia geral é utilizada nos métodos descritos a seguir.

#### A. Método dos Contadores Dedicados

Um TS MPEG-2 pode conter mais de um programa, inclusive com diferentes bases de tempo [1]; esse fluxo multiplexado de dados é comumente chamado de Fluxo de Transporte com Múltiplos Programas (*Multiple Program Transport Stream - MPTS*). Devido a isso, uma solução intuitiva para o problema da correção de PCR seria utilizar um número de contadores de 27MHz igual ao número de bases de tempo diferentes [2].

O primeiro passo para esta abordagem é identificar os pacotes que contêm informação de PCR e extrair a mesma. Os dados resultantes são então carregados no registro do contador referente ao programa, no próximo gatilho do relógio de 27MHz, incrementado por 1 ( $PCR+1$ ). Logo que a adaptação de taxa é concluída e o respectivo pacote enviado pelo sistema, o conteúdo do contador alocado é recuperado e usado para substituir o PCR original.

A principal desvantagem deste método é que muitos contadores podem ser necessários (e.g. muitos recursos de FPGA usados para criar os geradores), dependendo do número de programas no TS de entrada. Além disso, cada contador permanecerá ocupado até que o respectivo pacote seja enviado pelo sistema.

#### B. O Método da Compensação

O método da compensação [3], [7] é baseado em um único contador de 27MHz, utilizado para corrigir a informação de PCR de todos os programas. Quando o pacote contendo PCR é detectado, o estado atual do contador ( $Count_{in}$ ) é tomado e subtraído do respectivo valor de PCR; o resultado ( $PCR_I$ ) é então usado para substituir o valor original. Quando o pacote é enviado pelo sistema, o PCR é mais uma vez alterado: o estado atual do contador ( $Count_{out}$ ) é novamente tomado e somado ao valor do campo de PCR. Dessa forma, cada PCR original é substituído por uma nova referência, dada por:

$$PCR_I(i) = PCR(i) - Count_{in}(i) \quad (2)$$

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

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

onde  $\Delta PCR_{comp}(i) = Count_{out}(i) - Count_{in}(i)$  é o termo de compensação para  $PCR(i)$ .

O método da compensação é capaz de corrigir os valores de PCR, no entanto, apresenta a seguinte desvantagem: várias adições de 42-bits, o que apresenta maior complexidade quando comparado ao método dos contadores dedicados (Subseção II-A). Além disso, dependendo da taxa de chegada de pacotes com PCR, é possível que as operações não sejam executadas em um tempo apropriado, caso um hardware adequado não seja utilizado, comprometendo o processo de correção [9].

### III. O MÉTODO PROPOSTO: UM CONTADOR CONTROLADO POR SEMÁFOROS

Pelo que foi exposto nas seções anteriores, os métodos tradicionais apresentam vantagens e desvantagens, e a escolha

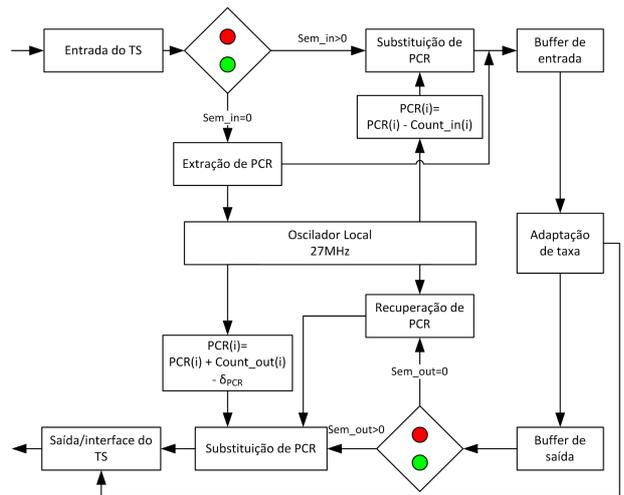


Fig. 1. O diagrama de blocos do método proposto: o contador controlado por semáforos.

de uma abordagem específica depende do que a aplicação final pode fornecer: poder de processamento ou recursos. Se os contadores dedicados forem usados, haverá uma grande quantidade de recursos alocados para o módulo de correção de PCR, no entanto, eles podem não ser sempre usados. Isto é devido a dois aspectos principais: é possível que diferentes programas usem a mesma base de tempo, e pacotes com PCR não estão normalmente próximos. Por outro lado, quando o método da compensação é utilizado, implica em um grande impacto na complexidade computacional.

Contudo, uma abordagem que tirasse vantagem do comportamento típico do fluxo de transporte e utilizasse um único contador com um número mínimo de operações aritméticas (ou até mesmo nenhuma operação), seria então a melhor opção. Esta é a inspiração para um novo método, que apresenta as principais vantagens dos dois esquemas tradicionais: apenas um contador de 27MHz é utilizado para computar os ajustes de informação de PCR (extraída do respectivo pacote de TS), como no método da compensação (ver seção II-B), e valores de PCR podem ser corrigidos sem nenhuma adição, sendo carregados diretamente no registro do contador, como no método dos contadores dedicados (ver seção II-A). O diagrama de blocos do método proposto está ilustrado na Figura 1 e será abordado nos próximos parágrafos.

De uma forma geral, os TSs normalmente encontrados em aplicações comerciais, utilizados em sistemas terrestres e via satélite, apresentam poucos pacotes com PCR próximos, e este fato pode ser explorado para se reduzir o número de operações aritméticas, quando há apenas um contador disponível no sistema. Com este fim, uma lógica auxiliar foi criada para controlar o acesso ao contador local de 42 bits, que apresenta agora dois estados: o estado livre, que é identificado quando o contador está disponível para ser carregado diretamente, com um valor de PCR extraído de um pacote de entrada, e o estado ocupado, que indica que o registro do contador já foi carregado com um valor de PCR. Neste último, em vez de usar um contador dedicado para compensar o atraso, o sistema cria um novo valor de PCR, que incorpora o estado atual do contador e serve como marcador para se calcular a referência de tempo corrigida.

A lógica criada para indicar o estado atual do contador e

fornecer o acesso compartilhado é baseada em uma estrutura de semáforos, como mostrado na Figura 1. Dois semáforos são utilizados: um para a entrada ( $Sem\_in$ ) e outro para a saída ( $Sem\_out$ ), ambos inicializados com 0. Sempre que um pacote contendo informação de PCR é enviado para o *buffer* de entrada,  $Sem\_in$  é incrementado; do mesmo modo, quando um pacote contendo PCR deixa o módulo de adaptação de taxa,  $Sem\_out$  é incrementado. Se  $Sem\_in$  e  $Sem\_out$  apresentarem o mesmo valor, o que é sempre avaliado quando uma operação sobre  $Sem\_out$  é finalizada, ambos recebem o valor 0. Isto significa que todos os pacotes com dados de PCR, que chegaram ao *buffer* de entrada, já foram enviados, e o contador se encontra livre.

Como exposto no último parágrafo, o contador de 27MHz se encontra no estado livre quando  $Sem\_in$  é igual a 0, o que significa que, quando um pacote contendo PCR for enviado ao *buffer* de entrada, seu valor de PCR poderá ser diretamente carregado no registro do contador. Em seguida,  $Sem\_in$  é incrementado, o que força o contador a entrar no estado ocupado ( $Sem\_in > 0$ ); a partir deste momento, os próximos valores de PCR serão substituídos por uma nova referência, dada pelo estado atual do contador, até que  $Sem\_in$  seja novamente 0. O novo valor de PCR é dado por

$$PCR(i) = PCR(i) - Count\_in(i), \quad (5)$$

onde  $PCR(i)$ , na direita, é o valor original do PCR e  $Count\_in(i)$  é o estado atual do contador, no momento da chegada do pacote.

Quando o primeiro pacote deixa o sistema,  $Sem\_out$  apresenta o valor 0, o que indica que o valor corrigido deve ser recuperado diretamente do registro do contador; isto é então seguido de um incremento o valor de  $Sem\_out$ . Quando os pacotes restantes, que chegaram enquanto o primeiro ainda ocupava o contador ( $Sem\_in$  maior que 0), estão para ser enviados, seus valores de PCR são novamente alterados, de acordo com

$$PCR(i) = PCR(i) + Count\_out(i) - \delta_{PCR}, \quad (6)$$

onde  $PCR(i)$ , na direita, é o valor de PCR modificado quando o pacote chegou,  $Count\_out(i)$  é o estado do atual do contador, quando o pacote é enviado, e  $\delta_{PCR}$  é uma constante baseada na taxa de dados, usada por todos os valores de PCR. Tal fator está também incorporado ao bloco *recuperação de PCR*, na Figura 1, para  $Sem\_out$  0.

Se os semáforos não alcançarem o mesmo valor, quando os pacotes restantes forem enviados, mesmo após o primeiro pacote já ter sido processado, o contador local permanecerá ocupado; os próximos pacotes de PCR a chegarem ainda serão corrigidos com referências intermediárias, já que não podem haver interferências no registro do contador, enquanto ainda houver pacotes com referências baseadas em seu registro.

A principal característica desta nova abordagem consiste no fato de que, se houver poucos pacotes com PCR próximos ou se estes estiverem muito afastados um do outro, então as operações aritméticas serão raras e o sistema poderá realizar todas as tarefas necessárias de maneira bastante rápida. Mesmo em um caso extremo, ao menos uma das correções, em uma sequência de pacotes muito próximos, não exigirá operações aritméticas.

O módulo de correção pode ser visto como duas máquinas independentes, com uma sendo executada na entrada e outra na saída do sistema. O pseudo-algoritmo para a entrada é:

- 1) Verificar  $Sem\_in$ ;
- 2) Se  $Sem\_in > 0$ , ir ao Passo 4;
- 3) Carregar  $PCR(i)$  no registro do contador e ir ao Passo 5;
- 4) Fazer  $PCR(i) = PCR(i) - Count\_in(i)$ ;
- 5) Incrementar  $Sem\_in$ ;
- 6) Fim.

O pseudo-algoritmo para a saída é:

- 1) Verificar  $Sem\_out$ ;
- 2) Se  $Sem\_out > 0$ , ir ao Passo 4;
- 3) Recuperar  $PCR(i)$  do registro do contador e ir ao Passo 5;
- 4) Fazer  $PCR(i) = PCR(i) + Count\_out(i) - \delta_{PCR}$ ;
- 5) Incrementar  $Sem\_out$ ;
- 6) Se  $Sem\_out \neq Sem\_in$ , ir ao Passo 8;
- 7) Fazer  $Sem\_out$  e  $Sem\_in$  iguais a 0;
- 8) Fim.

Dado que outras referências do TS são dependentes do PCR, como Referência de Tempo de Apresentação (*Presentation Time Stamp* - PTS) e Referência de Tempo de Decodificação (*Decoding Time Stamp* - DTS),  $\delta_{PCR}$  é importante para evitar quebras na cadeia de sincronização, causadas por grandes deslocamento no valor de PCR. Como já foi mencionado,  $\delta_{PCR}$  é dependente da taxa de dados e pode ser considerado proporcional ao atraso sofrido pelos pacotes com PCR, durante o processo de correção. No entanto, também é dependente da estratégia de *buffer*, isto é, a maneira como os pacotes são organizados para serem transmitidos. Por exemplo, uma estratégia bastante simples é assegurar que há ao menos  $N$  pacotes armazenados antes de se realizar a transmissão. Num caso extremo, se apenas um pacote com PCR chegar pela interface de entrada, pacotes nulos serão enviados, até que outros  $N - 1$  pacotes forem armazenados no *buffer*. O tempo decorrido, neste caso, é dado por:

$$IPT = \frac{188 * (N - 1)}{IN\_RATE}, \quad (7)$$

onde 188 é o tamanho, em bytes, de um pacote MPEG-2,  $IPT$  é o tempo de entrada de pacotes (*Input Packets Time*), em segundos, e  $IN\_RATE$  é a taxa de transporte da entrada, em bytes/segundo. Assim que todos os pacotes necessários tiverem chegado, o primeiro pacote é então enviado. No entanto, se os novos pacotes terminaram de chegar justamente no início da transmissão de um novo pacote de saída, pelo menos este processo será finalizado, antes que outros pacotes sejam enviados. O atraso adicional é dado por

$$OPT = \frac{188}{OUT\_RATE}, \quad (8)$$

onde  $OPT$  é o tempo de saída de pacote (*Output Packet Time*), em segundos, e  $OUT\_RATE$  é a taxa de transporte da saída, em bytes/segundo. O intervalo de tempo de correção total, em média, pode ser dado pelas Equações 7 e 8, como

$$CTI_{exp} = IPT + OPT, \quad (9)$$

onde  $CTI_{exp}$  é o intervalo de tempo de correção total esperado (*Expected Correction Time Interval*).

Baseado no que foi apresentado e que  $\delta_{PCR}$  deve ser expressado em gateios de 27MHz, é possível obter, da Equação 9, a expressão

$$\delta_{PCR} = \lfloor \alpha * CTI_{exp} * SYS\_CLK\_FREQ \rfloor, \quad (10)$$

onde  $SYS\_CLK\_FREQ$  é 27 MHz e  $\alpha$  é uma constante de escalonamento que serve para não se recuar demais os valores de PCR, sendo a sua escolha empírica.

## IV. DISCUSSÃO

O esquema proposto pode ser visto como um esquema completo de correção de PCR em MPTSs, fornecendo uma máquina de ajustes de referências de tempo e uma estrutura de acesso compartilhado. Quando comparado aos métodos tradicionais, esta nova metodologia possibilita uma implementação mais leve e compacta, dado que apenas um contador é usado e o número de operações aritméticas é consideravelmente reduzido.

A quantidade de recursos usados em cada esquema é variável e depende da implementação. Mesmo na literatura disponível, onde se encontram informações sobre os recursos necessários ([7], [9], [10]), o número apresentado é bastante variável. No entanto, nas experiências realizadas durante o presente trabalho, a estrutura de controle baseada em semáforos tomou apenas poucas unidades lógicas, o que comprometeria menos do que o necessário para um somador de 42 bits ou um contador de 27MHz. Assim, é possível prever que o método proposto, na grande maioria de suas aplicações, exigirá menos células lógicas que o método da compensação [7], com dois ou mais somadores, e que o método dos contadores dedicados [9], quando forem empregados mais de dois contadores.

Com relação ao número de operações aritméticas exigidas para a correção de PCR, é possível estimar um valor esperado. Por exemplo, o método da compensação realiza duas operações para cada valor de PCR, o que leva a:

$$NAP_{comp} = 2 \sum_{i=0}^{N-1} \left( \frac{1}{PCR\_RATE(i)} \right), \quad (11)$$

onde  $NAP_{comp}$  é o número de operações aritméticas, por segundo, no método da compensação,  $N$  é o número de bases de tempo e  $PCR\_RATE(i)$  é a taxa de chegada de valores de PCR de uma dada base de tempo  $i$ , em segundos. Para o contador controlado por semáforos, uma formulação matemática não é tão simples, dado que o número de operações aritméticas depende das taxas de PCRs, da proximidade no TS, das diferentes bases de tempo, e do atraso nos pacotes de PCR.

A taxa de chegada de pacotes de PCR, para cada base de tempo, pode ser vista como um evento probabilístico independente, que é bem sucedido se uma referência chega durante o intervalo de correção, com uma probabilidade dependente da taxa de PCR. Se mais de uma referência chegar ao módulo de correção de PCR durante este intervalo, operações aritméticas serão realizadas. Além disso, se um pacote com PCR chegar no final do intervalo de correção, a próxima referência, no próximo intervalo, resultará novamente em operações aritméticas. Sendo assim, o intervalo de tempo, durante o qual a chegada de novas referências resultam em operações, será considerado  $2CTI_{exp}$ . Supondo que há  $N$  diferentes bases de tempo e  $2(n-1)$  operações ocorrem quando  $n > 1$  de pacotes chegam durante o intervalo de tempo considerado, o número esperado de operações é dado por

$$NAP_{sem} = \sum_{n=2}^N \frac{(n-1)}{CTI_{exp}} \sum_{A \in F_n} \prod_{l \in A} p_l \prod_{k \in A^c} (1-p_k), \quad (12)$$

onde  $NAP_{sem}$  é o número de operações aritméticas, por segundo, no método do contador controlado por semáforos,  $N$  é o número de diferentes bases de tempo,  $F_n$  é o conjunto de todos os subconjuntos de  $n$  bases de tempo, que podem chegar durante o intervalo  $2CTI_{exp}$ ,  $p_i$  é a probabilidade

de uma referência pertencente à base de tempo  $i$  chegar durante o mesmo intervalo, que pode ser aproximado por  $2CTI_{exp}/PCR\_RATE(i)$ , e  $A^c$  é o complemento de  $A$ .

Como exemplo, um TS com 8 bases de tempo diferentes, no qual duas delas chegam com uma taxa de 24 ms e as outras com 33 ms, que é convertido de uma taxa de 24.1Mbps para 43Mbps, com pelo menos dois pacotes armazenados em *buffer* (a mesma configuração apresentada nas simulações na próxima seção), resultaria em um  $NAP_{comp} = 530,3030$  e um  $NAP_{sem} = 11,7913$  operações por segundo. Isto mostra que apenas 2,22% das operações necessárias, no método da compensação, seriam exigidas no método do contador controlado por semáforos.

## V. RESULTADOS DE SIMULAÇÃO

Para se avaliar o desempenho do método proposto, simulações em dois tipos de cenários foram realizadas. No primeiro, 30 segundos de um TS com 14 programas e 8 diferentes bases de tempo foram processados por um conversor de taxa, usando o contador controlado por semáforos para corrigir as informações de PCR. A taxa de bits multiplexada é de aproximadamente 24,1Mbps, e os resultados referentes aos dois primeiros programas são mostrados na Figura 2, de modo a se avaliar o *jitter* na saída do sistema. As taxas de bits dos programas 1 e 2 são 1,85Mbps e 2,17Mbps, respectivamente. O fluxo de saída apresenta uma taxa de 43 Mbps, que é alcançada, no módulo de adaptação de taxa, pela introdução de pacotes nulos no TS original.

O módulo de adaptação de taxa é composto pelos *buffers* de entrada e saída e um algoritmo de escalonamento. Os pacotes de entrada são armazenados no *buffer* de entrada e, sempre que um pacote é enviado para o *buffer* de saída, o módulo de adaptação de taxa verifica se há ao menos dois pacotes. Neste caso, o primeiro pacote armazenado é diretamente copiado para o *buffer* de saída; caso contrário, um pacote nulo é enviado. Pacotes nulos no TS de entrada são sempre ignorados e criados unicamente na saída, pelo adaptador de taxa. O valor  $\alpha = 0,5$  foi adotado para  $\delta_{PCR}$ , que então foi fixado em 1314.

Durante a simulação usando o MPTS apresentado, a evolução dos parâmetros dos semáforos foi traçada e analisada. Observou-se que o valor máximo do semáforo de entrada foi 3, indicando que, no pior caso, apenas 2 operações aritméticas foram executadas em paralelo. Além disso, 97,9% dos dados de PCR foram diretamente carregados no registro do contador, sendo o resto corrigido por referências intermediárias, tiradas do contador. Assim, o registro do contador está quase sempre disponível, o que também pode ser constatado para a maioria dos TSs comerciais.

Devido ao método proposto, a correção de PCR realizada pelo conversor de taxa executou apenas 330 operações (de 42 bits), em vez das 15836 exigidas pelo método da compensação. Além disso, apenas um contador de 27MHz foi utilizado, ao contrário dos 8 contadores que seriam requeridos pelo método dos contadores dedicados, ou dos 3 necessários ao método em [9]. Dessa forma, a complexidade computacional apresentada pelo método proposto é muito menor que a apresentada pelo método da compensação, e os requisitos de hardware podem ser aliviados, quando comparado ao método dos contadores dedicados.

Na Figura 2, percebe-se que o *jitter* de saída é estável, muito próximo do que é apresentado na entrada do sistema e menor que o máximo aceitável de 500ns [1]. Isto mostra que o

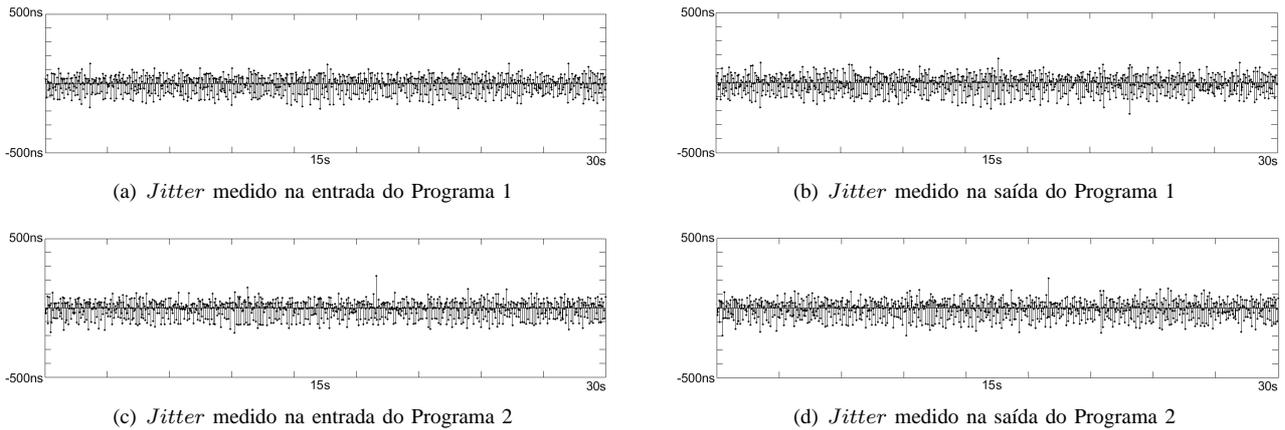


Fig. 2. Resultados de simulação referentes aos três primeiros programas do TS.

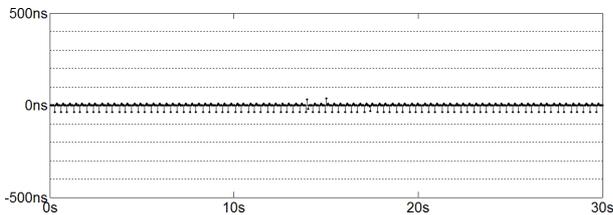


Fig. 3. Jitter relacionado à correção de PCR para programa com jitter de entrada zero.

método proposto é eficiente e não compromete a sincronização do sistema.

No segundo cenário, 30 segundos de um TS com uma taxa de bits de 32,507936 Mbps e dois programas, com jitter de PCR zero, são processados por um conversor de taxa, utilizando o método do contador controlado por semáforos para a correção de informação de PCR. O primeiro programa apresenta uma taxa de 15,52 Mbps, com vídeo em alta definição, e o segundo contém vídeo em baixa definição, a uma taxa menor que 1,0 Mbps. Os resultados referentes ao primeiro programa são mostrados na Figura 3, de modo a se avaliar o jitter inserido pelo método proposto. A taxa de saída é novamente fixada em 43 Mbps.

Os resultados apresentados por outros métodos não foram obtidos com as mesmas condições e TSs, no entanto, uma comparação simplificada é possível. A Tabela I mostra o jitter de saída do método proposto e também os de outros existentes na literatura, mostrando que o desempenho deste é superior aos apresentados pelos demais esquemas.

TABELA I  
COMPARAÇÃO COM OUTROS MÉTODOS

Método	Jitter de saída
Método proposto	< 50ns
Xingdong et al. [3]	≈ 108ns
Chen et al. [10]	74ns
He et al. [4]	> 100ns

## VI. CONCLUSÕES

Um novo esquema, baseado em um contador controlado por semáforos, foi apresentado, com o objetivo de compensar o jitter causado pelo módulo de adaptação de taxa, em um

processador de TSs MPEG-2. O novo método se mostrou eficaz, no sentido de que mantém o jitter de saída abaixo dos limites recomendados e próximo ao medido na entrada. Além disso, ele emprega uma configuração de hardware mínima para MPTSs com diferentes bases de tempo, requer menor poder de processamento e adiciona menos jitter, quando comparado a métodos presentes na literatura. Deve-se notar que a lógica adicional para a implementação da estrutura de semáforos é simples e fácil de ser sintetizada.

## AGRADECIMENTOS

Os autores gostariam de agradecer ao CT-PIM e à UNISOL, pelo suporte fornecido para a realização deste trabalho.

## REFERÊNCIAS

- [1] ISO/IEC 13818-1: Information technology - Generic coding of moving pictures and associated audio information - Part 1: Systems. 3rd ed. Geneva, Switzerland: 2007.
- [2] 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.
- [3] 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.
- [4] Y. He, J. Zhou, and Y. Zhou, "Implementation of TS De-multiplexer with FPGA in DVB-IP Gateway for Network TV", *IEEE 7th Workshop on Multimedia Signal Processing*, Shanghai, China, October 2005.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] W. Fischer, "Digital Video and Audio Broadcasting Technology: A Practical Engineering Guide, 3rd ed., Berlin, Germany: Springer, 2010.
- [9] 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.
- [10] 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.