

Multipleksowanie strumieni MPEG-2

Ćwiczenie dotyczy multipleksacji wejściowych strumieni elementarnych, jednego wizyjnego oraz jednego fonicznego, w strumień programowy zdefiniowany w standardzie MPEG-2 (ISO/IEC 13818-1). Multipleksacja jest dokonywana przez program o nazwie **mpsc** przygotowany przez Pana Tomasza Dzieścielewskiego w ramach projektu inżynierskiego. Wszelkie prawa autorskie są zastrzeżone.

Wprowadzenie:

Celem ćwiczenia jest zapoznanie się z koderem warstwy systemowej standardu MPEG-2. Ćwiczenie polega na multipleksacji wejściowych strumieni elementarnych, jednego wizyjnego oraz jednego fonicznego, w strumień programowy zdefiniowany w standardzie MPEG-2 (ISO/IEC 13818-1).

Przyjęte zostały następujące założenia dotyczące elementarnych strumieni wejściowych:

- Strumień wizyjny – strumień MPEG-2 Obraz (ISO/IEC 13818-2) (strumień zgodne z MPEG-1 ISO/IEC IS 11172-2 mogą nie być poprawnie interpretowane) o stałej prędkości (CBR); profil główny, poziom główny; sekwencje progresywne; low_delay=0.
- Strumień foniczny – zgodne z IS/IEC 11172-3 (rozszerzenia wielokanałowe i niższych przepływności opisane w IS/IEC 13818-3 nie są uwzględnione w ćwiczeniu) o stałej prędkości (CBR); warstwy: I, II.

Wyjściowy strumień programowy jest strumieniem o stałej prędkości.

Opis teoretyczny zagadnienia:

2. Opis teoretyczny

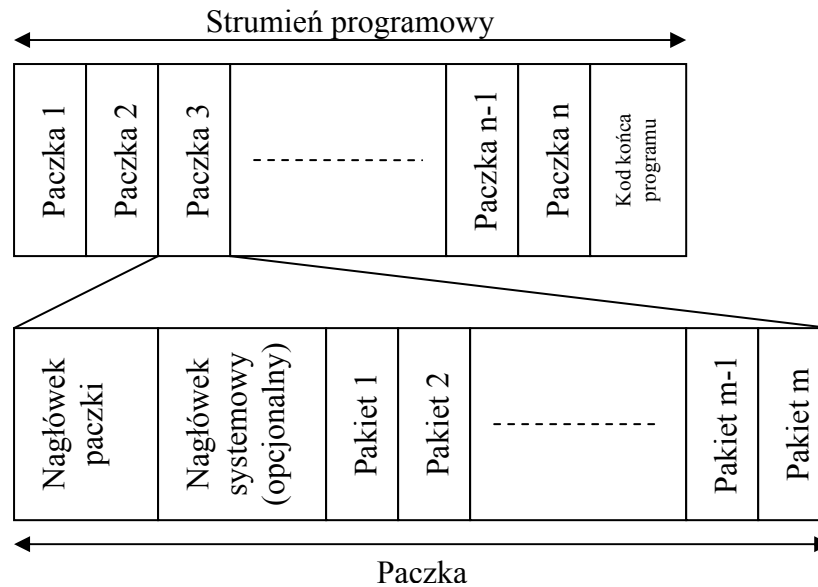
2.1. Budowa strumienia programowego.

Ogólna struktura strumienia programowego została przedstawiona na rys. 1.

Strumień programowy jest strukturą łączącą wiele strumieni elementarnych (ES) w jeden program.

Dane należące do ES przenoszone są w strumieniu programowym w postaci pakietów PES, których długość, w ogólności, nie jest ustalona, choć w szczególności może być stała (wykorzystywany w ćwiczeniu koder tworzy pakiety o stałej długości).

Pakiety PES grupowane są w paczkach. Warstwa paczki spełnia głównie funkcje synchronizacyjne, co zostanie opisane w dalszej części niniejszego dokumentu. Ponadto, w paczce przesyłane są również inne, poza pakietami strumieni elementarnych (audio, wideo, prywatnych i wypełniającego), struktury, takie jak nagłówki systemowy, mapa programu czy kartoteka strumienia programowego. Dwie ostatnie z nich przenoszone są w postaci pakietów o specjalnym formacie.



Rys. 1 – Struktura strumienia programowego.

2.2. Synchronizacja strumieni elementarnych przenoszonych w strumieniu programowym.

Podstawowym problemem związanym z multipleksacją kilku strumieni jest zapewnienie ich synchronizacji w dekodерze. Aby umożliwić realizację tego zadania, stosowane są dwa znaczniki (stemple) czasowe – PTS i DTS.

Zanim przejdziemy do omówienia ich funkcji, konieczne jest zdefiniowanie dwóch pojęć: jednostki prezentacji PU oraz jednostki dostępu AU.

Za jednostkę prezentacji, mówiąc w uproszczeniu, możemy uważać obraz – w przypadku strumienia wizyjnego – lub ramkę (o pewnej stałej, zależnej od warstwy, w której pracuje koder foniczny, liczbie próbek) – w przypadku strumienia fonicznego. Dokładne definicje obu pojęć podane są w [1], [2] i [3].

Znacznik PTS przyporządkowany jest jednostce prezentacji i wyznacza moment, w którym powinna ona zostać przedstawiona odbiorcy. Znacznik DTS odnosi się do jednostki dostępu i określa chwilę, w której powinno rozpocząć się jej dekodowanie do jednostki prezentacji.

Jednostką czasu, w której kodowane są oba znaczniki jest okres zegara o częstotliwości 90kHz. Stanowi on trzystukrotność okresu zegara systemowego, którego częstotliwość ustalono na 27MHz.

Znaczniki PTS i DTS kodowane są w nagłówku pakietu i odnoszą się do pierwszej jednostki prezentacji lub dostępu rozpoczynającej się w danym pakiecie. Należy zauważyć, iż konieczność stosowania dwóch różnych znaczników dla czasu prezentacji i dekodowania wynika wyłącznie z faktu, iż dla poszczególnych obrazów strumieni wizyjnych, czasy te mogą różnić się od siebie, co jest konsekwencją kodowania obrazów typu B. Ramki foniczne mają zawsze identyczne czasy dekodowania i prezentacji, dlatego w pakietach audio koduje się tylko jeden znacznik – PTS. W pakietach wizyjnych koduje się oba znaczniki, gdy są one różne, w przeciwnym razie można kodować wyłącznie znacznik PTS lub też oba.

Podstawa czasu kodera kodowana jest w warstwie paczki, w polach SCR nagłówka paczki. Pola te zawierają reprezentację próbki zegara systemowego kodera o częstotliwości, jak wspomniano wyżej, równej 27MHz. Służą one do

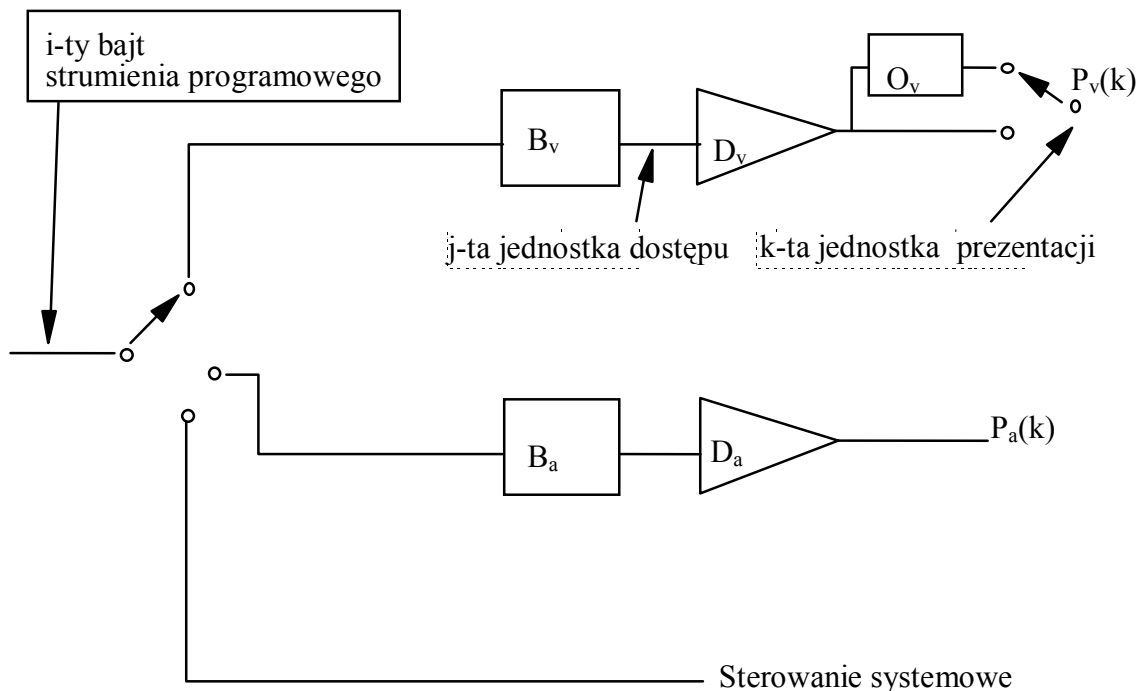
zsynchronizowania zegara dekodera z zegarem koder, co najczęściej realizowane jest za pomocą pętli PLL.

2.3. Model dekodera systemowego P-STD.

P-STD jest matematycznym modelem dekodera systemowego dla strumienia programowego określonym w ISO/IEC 13818-1 [1]. Celowość zdefiniowania takiego modelu jest uzasadniona koniecznością zapewnienia pewnej jednorodności strumieni wytwarzanych przez rzeczywiste kodery. Każdy strumień programowy wytwarzany przez dowolną implementację koder systemowego MPEG-2, musi być poprawnie dekodowany przez dekodera P-STD.

Należy wyraźnie zaznaczyć, że model P-STD nie narzuca sposobu fizycznej realizacji rzeczywistego dekodera, lecz służy raczej do określenia pewnych ograniczeń nałożonych na sam strumień.

Poniższy rysunek przedstawia model dekodera P-STD dla strumienia programowego wytwarzanego przez projektowany koder. Oznaczenia przyjęte na rysunku różnią się od oznaczeń stosowanych w [1].



Rys. 2 – Model dekodera P-STD dla strumienia programowego wytwarzanego przez projektowany koder.

Bufory B_v (o rozmiarze BS_v) oraz B_a (o rozmiarze BS_a) są buforami wejściowymi dla strumienia, odpowiednio, wizyjnego oraz fonicznego. Umieszczane są w nich dane pochodzące z pakietów PES odpowiednich strumieni elementarnych. Dane systemowe nie są umieszczane w żadnym z buforów, lecz mogą być wykorzystywane do sterowania pracą systemu.

D_v oraz D_a są dekoderni jednostek dostępu wideo i audio. Proces dekodowania rozpoczyna się w chwili określonej przez znacznik PTS lub DTS lub też, jeśli dana jednostka dostępu nie ma przyporządkowanego PTS (DTS), w chwili wyznaczonej na

podstawie informacji zawartej w strumieniu elementarnym. W modelu P-STD przyjmuje się, iż czas dekodowania jednostki dostępu jest równy zero, tj. dekodowanie zachodzi w jednej chwili. Bajty dekodowanej, przebywającej najdłużej w buforze, jednostki dostępu usuwane są z bufora wejściowego, jednostka dostępu zostaje zaś zdekodowana do jednostki prezentacji, która, w przypadku strumienia fonicznego, prezentowana jest natychmiast użytkownikowi. W przypadku strumienia wizyjnego, jednostka prezentacji może wpieryw zostać umieszczona w buforze zmiany porządku O_v , w którym następuje szeregowanie obrazów według kolejności ich prezentacji.

Z punktu widzenia poprawnej realizacji kodera, niezwykle ważny jest warunek dotyczący pracy buforów wejściowych. Strumień programowy powinien być tworzony w taki sposób, by bufor te nie ulegały przepełnieniu ani opróżnieniu. Istnieją wyjątki od tej zasady, lecz wykraczają one poza zakres ćwiczenia i jako takie nie będą omawiane.

2.4. Struktury występujące w strumieniu programowym.

Wykorzystywana implementacja kodera używa takie struktury jak:

- Paczka (nagłówek paczki)
- Nagłówek systemowy
- Mapa programu
- Pakiet PES strumienia elementarnego audio i wideo
- Pakiet PES strumienia wypełniającego

3. Algorytm multipleksacji.

3.1. Wprowadzenie.

Zastosowany w projekcie algorytm multipleksacji strumieni elementarnych jest rozwinięciem przykładu opisanego w [4].

Założenie stałej prędkości, tak strumieni wejściowych, jak i strumienia wynikowego, a także stałego rozmiaru pakietu PES, pozwala na deterministyczne wyznaczenie schematu przepływu pakietów wideo i audio. W efekcie otrzymujemy strukturę strumienia programowego analogiczną do zwielokrotnienia TDM, gdzie każdy strumień elementarny ma przyporządkowany „kanał” w strumieniu programowym. Fakt ten implikuje konieczność zastosowania strumienia wypełniającego odpowiednie „szczeliny” w sytuacji, gdy na wejściu kodera nie znajduje się strumień elementarny, któremu „szczeliny” te są przyporządkowane. Także w przypadku zakończenia się któregoś ze strumieni elementarnych, należy wypełnić lukę powstałą w ostatnim jego pakiecie poprzez użycie bądź to bajtów wypełniających nagłówek pakietu, lub też za pomocą pakietu strumienia wypełniającego.

Strumień programowy podzielony jest na paczki, zawierające oprócz pakietów strumieni elementarnych, także inne struktury, takie jak nagłówek paczki, nagłówek systemowy, czy mapa programu. Aby zachować regularność budowy strumienia programowego, konieczne jest użycie strumienia wypełniającego w miejscu odpowiednich struktur, gdy te nie są przesyłane. Ponadto zakłada się, iż w każdej paczce przenoszona jest stała liczba pakietów PES.

3.2. Wyznaczanie schematu multipleksacji.

Stosunek prędkości strumienia wideo do prędkości strumienia audio wynosi:

$$rate_ratio = video_rate / audio_rate \quad (1)$$

Wyznacza on sposób przepływu pakietów wideo i audio zgodnie z następującą regułą:

Niech:

$$rate_ratio = rate_ratio_I + rate_ratio_F \quad (2)$$

gdzie:

$rate_ratio_I$ – część całkowita $rate_ratio$

$rate_ratio_F$ – część ułamkowa $rate_ratio$

Niech n będzie zaokrągloną do jedności wartością $(1 / rate_ratio_F)$.

Niech $k = rate_ratio_I$.

Pakiet audio jest przeplatany k -toma pakietami wideo, z wyjątkiem każdego n -tego pakietu audio, przed którym występuje $k+1$ pakietów wideo.

3.3. Prędkość strumienia programowego.

Prędkość łącznego strumienia danych audio i wideo:

$$data_rate = video_rate + audio_rate \quad [\text{bajtów/s}] \quad (3)$$

gdzie wszystkie prędkości podane są w bajtach/s.

Prędkość strumienia programowego:

$$mux_rate = data_rate * (1 + (packet_header_size + pack_overhead / packets_in_pack) / packet_data_size) \quad [\text{bajtów/s}] \quad (4)$$

gdzie:

$$pack_overhead = pack_header_size + system_header_size + program_stream_map_size$$

$packet_header_size$ – stały rozmiar nagłówka pakietu w bajtach

$pack_header_size$ – stały rozmiar nagłówka paczki w bajtach

$packet_data_size$ – stały rozmiar ładunku pakietu w bajtach

$system_header_size$ – rozmiar nagłówka systemowego w bajtach

$program_stream_map_size$ – rozmiar pakietu zawierającego mapę programu w bajtach

$packets_in_pack$ – stała liczba pakietów w paczce

Przyjęto następujące założenia dotyczące budowy strumienia programowego:

- Nagłówek systemowy przesyłany jest wyłącznie w pierwszej paczce.
- Mapa programu przesyłana jest wyłącznie w pierwszej paczce w postaci jednego pakietu.
- W kolejnych paczkach, brak w/w struktur jest kompensowany za pomocą strumienia wypełniającego, co w efekcie pozwala uzyskać strumień programowy o stałej prędkości równej mux_rate .

3.4. Próbkki zegara systemowego.

Próbkki zegara systemowego zapisywane są w nagłówkach kolejnych paczek i obliczane są według poniższego równania:

$$SCR [n] = SCR [1] + (n - 1) * pack_size * mux_rate * 27 * 10^6 [T_{27\text{MHz}}] \quad (3)$$

gdzie:

n – nr paczki

$SCR[n]$ – wartość SCR w n-tej paczce

$pack_size$ – stały rozmiar paczki w bajtach

$$SCR [1] = 9 * mux_rate * 27 * 10^6 [T_{27\text{MHz}}] \quad (4)$$

Należy zauważyć, iż przyjęte założenia dotyczące budowy strumienia programowego, zapewniają spełnienie warunku stałego rozmiaru paczki.

3.5. Wyznaczanie momentu wejścia strumieni elementarnych w strumień programowy.

3.5.1. Opis problemu.

Wykorzystywany koder pozwala na wzajemne przesunięcie strumieni wideo i audio, rozumiane jako opóźnienie rozpoczęcia prezentacji jednego bądź obu z nich. Ponadto, należy zauważyć, iż jednostki dostępu wideo mają znacznie większy rozmiar niż jednostki dostępu audio, co sprawia, że na jedną AU strumienia wideo może, w strumieniu programowym, przypadać wiele AU strumienia audio. Oba te czynniki, w świetle ograniczonych rozmiarów buforów dekodera P-STD, stwarzają możliwość zaistnienia przepełnienia któregoś z nich, co miałyby miejsce, gdyby rozpoczęcie transmisji któregoś ze strumieni elementarnych nastąpiło zbyt wcześnie w stosunku do planowanego czasu jego prezentacji.

3.5.2. Rozwiązanie problemu.

W celu uniknięcia wspomnianej sytuacji, przyjęto, iż każdy strumień elementarny zostanie umieszczony w strumieniu programowym począwszy od ostatniej możliwej pozycji („szczeliny”), gwarantującej, iż w odpowiednim buforze dekodera P-STD, przed rozpoczęciem dekodowania pierwszej jednostki dostępu (którego moment określony jest przez przyporządkowany jednostce znacznik DTS), znajdą się wszystkie bajty do niej należące. Wszystkie poprzedzające tę pozycję „szczeliny” zostaną wypełnione pakietami strumienia wypełniającego.

3.5.3. Algorytm wyznaczania momentu wejścia strumieni elementarnych w strumień programowy.

Algorytm wyznaczania momentu wejścia strumieni elementarnych w strumień programowy został przedstawiony na rys. 4.

Definicje:

- Szczelina wideo – miejsce w strumieniu programowym, w którym, w świetle wyznaczonego schematu multipleksacji, powinien być transmitowany pakiet PES strumienia wideo.
- Szczelina audio - miejsce w strumieniu programowym, w którym, w świetle wyznaczonego schematu multipleksacji, powinien być transmitowany pakiet PES strumienia audio.
- T_a – czas planowego rozpoczęcia dekodowania strumienia audio.
- T_v – czas planowego rozpoczęcia dekodowania strumienia wideo.
- T_{delay} – zadane globalne opóźnienie czasu dekodowania obu rodzajów strumieni.

Parametry T_a , T_v oraz T_{delay} są ustalane przez użytkownika, przy czym opóźnienie T_{delay} może zostać zwiększone w wyniku realizacji poniższego algorytmu.

Po wykonaniu przedstawionego algorytmu, zmienne start_a oraz start_v określają numery szczelin, od których należy rozpocząć transmisję odpowiednio strumienia audio i wideo.

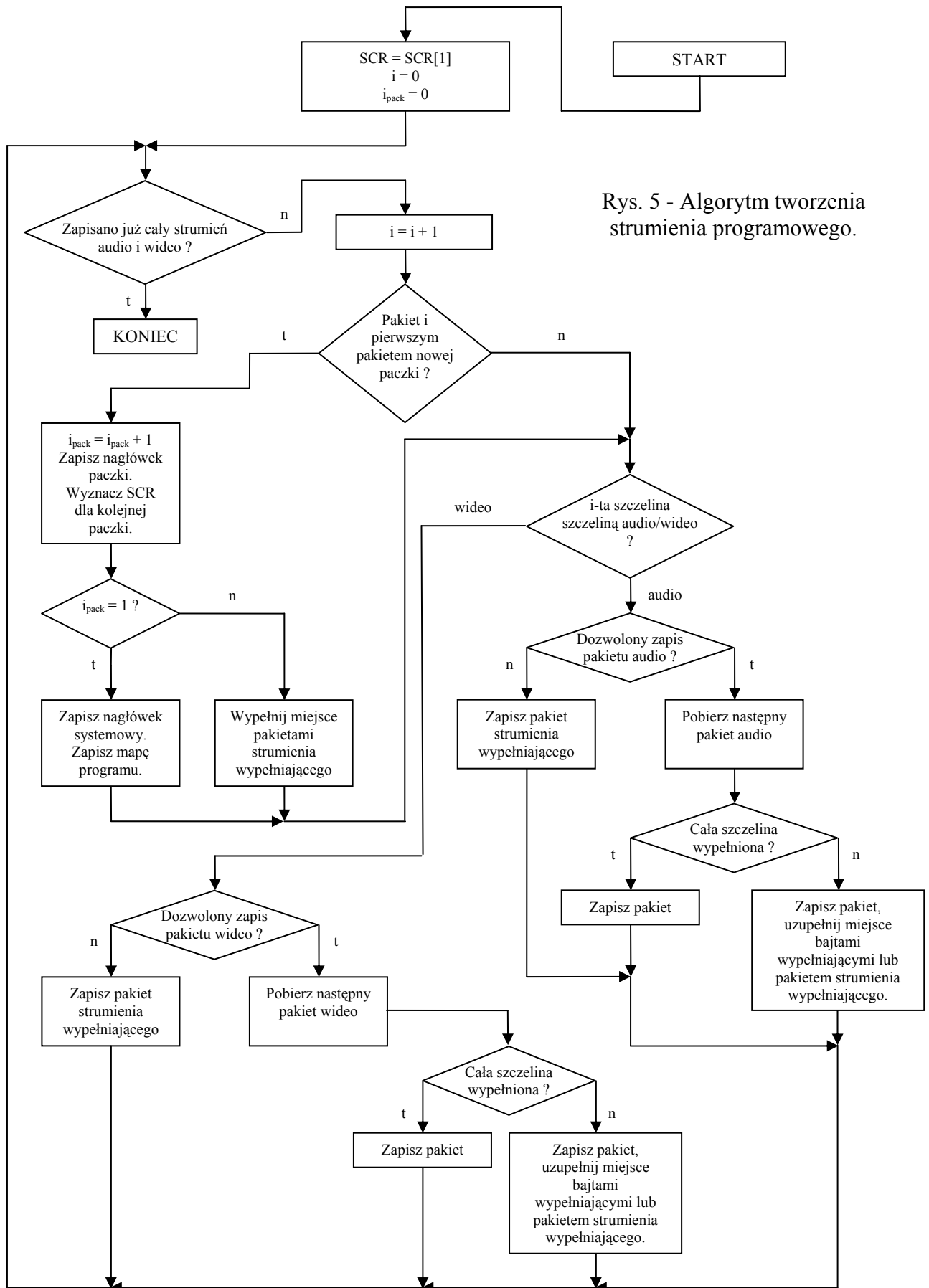
3.6. Algorytm tworzenia strumienia programowego.

Rys. 5 przedstawia algorytm tworzenia strumienia programowego.

Komentarza wymaga blok „dozwolony zapis pakietu wideo (audio)?”. Decyzja o zgodzie na zapis bądź też jej odmowie polega na sprawdzeniu, czy strumień wejściowy nie skończył się oraz na porównaniu numeru i aktualnego pakietu (szczeliny) z numerem szczeliny, od której strumień elementarny powinien być zapisywany w strumieniu programowym (tj. $start_v$ lub $start_a$ z punktu 3.5.3).

Blok „pobierz kolejny pakiet wideo (audio)” obejmuje operacje wyjęcia ze strumienia elementarnego kolejnego pakietu oraz ustalenie jego znaczników PTS i DTS. Zapis znaczników PTS i DTS w nagłówku pakietu odbywa się zawsze, gdy jest to możliwe, tzn. gdy w pakiecie rozpoczyna się kolejna jednostka dostępu strumienia elementarnego.

W sytuacji, gdy zapisywany jest ostatni pakiet danego strumienia wejściowego, może zajść sytuacja, w której jego długość jest mniejsza od ustalonej długości pakietu PES. Wówczas koder uzupełnia szczelinę za pomocą pakietu strumienia wypełniającego, bądź też, jeśli liczba brakujących bajtów jest mała, za pomocą bajtów wypełniających nagłówka zapisywanego pakietu PES.



Rys. 5 - Algorytm tworzenia strumienia programowego.

3.7. Odczyt właściwości strumieni elementarnych.

Do poprawnej pracy, koder warstwy systemowej potrzebuje pewnych informacji dotyczących strumieni elementarnych, które mają podlegać multipleksacji. Tabela 2 zawiera wykaz właściwości strumieni elementarnych, których znajomość jest niezbędna do realizacji kodera systemowego, wraz z opisem ich przeznaczenia.

Typ strumienia	Właściwość	Przeznaczenie
Wideo	Prędkość strumienia	Wyznaczenie prędkości strumienia programowego
	Częstotliwość obrazów	Wyznaczenie znaczników PTS i DTS
	Typ obrazu	
	Rozmiar bufora VBV	Wyznaczenie rozmiaru bufora wejściowego dla strumienia wideo
Audio	Prędkość strumienia	Wyznaczenie prędkości strumienia programowego
	Częstotliwość próbkowania	Wyznaczenie znaczników PTS
	Liczba próbek w ramce	

Tabela 2 – Wykaz właściwości strumieni elementarnych, których znajomość jest niezbędna do realizacji kodera systemowego, wraz z opisem ich przeznaczenia.

4. Bibliografia

- [1] *Pr PN-EN ISO/IEC 13818-1: Ogólne zasady kodowania obrazów ruchomych i towarzyszącej im informacji dźwiękowej: Systemy.*
- [2] *ISO/IEC 13818-2: Generic coding of moving pictures and associated audio information: Video.*
- [3] *ISO/IEC IS 11172-3: Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s. Part 3: Audio.*
- [4] *ISO/IEC IS 11172-1: Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s. Part 1: Systems. Annex A.*

Polecenia:

1. Multipleksowanie strumieni

Przed przystąpieniem do multipleksacji proszę zapoznać się ze strumieniami elementarnymi wykorzystywanymi w ćwiczeniu (rozmiar strumieni, jakość, metoda kompresji).

Dokonać kilku multipleksacji wejściowych strumieni elementarnych, jednego wizyjnego oraz jednego fonicznego (warstwy I i II), w strumień programowy zdefiniowany w standardzie MPEG-2 (ISO/IEC 13818-1). Multipleksację dokonać z wykorzystaniem programu **mpsc**.

```
mpsc -v video\x.m2v -a "audio\Layer II\xx.mp2" -o program\xxx.mpg
```

Zwrócić uwagę na rozmiar strumienia wyjściowego, określić ilość informacji programowej towarzyszącej strumieniom elementarnym.

2. Analiza składni strumienia programowego – program BBINFO.

Do sprawdzenia zgodności tworzonego przez koder strumienia programowego z normą proszę wykorzystać program BBINFO. Program ten, oprócz prezentacji pól poszczególnych nagłówków strumienia, interpretuje ich wartości, informując o ewentualnych błędach.

Po upewnieniu się, że nagłówki występujących w strumieniu struktur tworzone są prawidłowo, proszę sprawdzić poprawność generowanej przez koder informacji synchronizacyjnej (pola DTS, PTS, SCR). Program BBINFO prezentuje wspomniane informacje w postaci czasowej, przez co sprawdzenie jej zgodności z założeniami staje się niezwykle proste. Ponadto, program ten analizuje stan buforów P-STD i informuje o ewentualnych nieprawidłowościach w ich pracy.

2. Odtwarzanie strumieni programowych.

Do odtwarzania generowanych przez koder strumieni programowych proszę użyć program Windows Media Player z kodekiem wideo – Elecard MPEG-2 Codec.

Zaprojektowany koder, ma zaimplementowany mechanizm sprawdzenia poprawności generowanej informacji synchronizacyjnej, w postaci możliwości przesunięcia momentu dekodowania strumieni elementarnych. Miało to umożliwić sprawdzenie zachowania się dekodowników w sytuacji, gdy na przykład nakazane zostanie rozpoczęcie odtwarzania strumienia audio po kilku sekundach od rozpoczęcia odtwarzania sekwencji wideo.

Proszę sprawdzić jak Microsoft Windows Player rozpoczyna dekodowanie przesuniętego strumienia w czasie. Czy istnieje pewna konsekwencja w działaniu dekodera, która pozwala stwierdzić, że odpowiedzialność za powstałą sytuację leży po stronie kodera?

Uwagi organizacyjne:

1. Katalogiem roboczym do umieszczania własnych plików jest **x:**. Przed przystąpieniem do ćwiczeń należy usunąć wszystkie pliki z tego katalogu.
 2. Katalog roboczy zawierający niezbędne pliki do wykonania ćwiczenia znajduje się na dysku **v:/dtv/mux/**.
 3. **Po zakończeniu ćwiczenia należy skasować wszystkie pliki z dysku x:\.**
-