

*Dr inż. Piotr Kawalec*

*Inż. Dariusz Koliński*

*Mgr inż. Jacek Mocki*

# **ZASTOSOWANIE PROGRAMOWALNYCH STRUKTUR LOGICZNYCH W URZĄDZENIACH STEROWANIA RUCHEM KOLEJOWYM**

## **SPIS TREŚCI**

1. Wstęp
2. Standardowe i specjalizowane układy cyfrowe
3. Pakiety wspomaganie komputerowego
4. Projektowanie układów sterowania w języku VHDL
5. Ocena wybranych parametrów układów srk zrealizowanych w różnych technologiach
6. Wnioski

## **STRESZCZENIE**

*Przedstawiono zagadnienia związane ze sprzętową realizacją bloków logicznych urządzeń srk w programowalnych strukturach logicznych. Zaprezentowano klasyfikację układów cyfrowych, ze szczególnym uwzględnieniem układów specjalizowanych, programowanych przez użytkownika. Opisano pakiety wspomaganie komputerowego, stosowanego do projektowania układów specjalizowanych w strukturach logicznych FPLD i przedstawiono procesy: specyfikacji, weryfikacji i implementacji projektowanych układów. Zaprezentowano stanowiska badawcze do prototypowania specjalizowanych układów sterowania, zrealizowanych w programowalnych strukturach logicznych. Na przykładzie bloku logicznego samoczynnej sygnalizacji przejazdowej oszacowano szybkość działania oraz wybrane parametry niezawodnościowe układów srk, zrealizowanych w różnych technologiach.*

# 1. WSTĘP

Rozproszenie geograficzne systemów sterowania ruchem kolejowym w połączeniu z przekazywaniem wstępnego przetwarzania danych oraz funkcji decyzyjnych na poziom lokalny powodują, że stawia się coraz większe wymagania w stosunku do lokalnych urządzeń sterujących w zakresie realizowanych algorytmów sterowania, przetwarzania informacji oraz pewności i szybkości transmisji danych. Dlatego też, od lat siedemdziesiątych XX w. w urządzeniach sterowania ruchem kolejowym (srk) coraz szerzej zaczęto stosować układy elektroniczne [1]. Budowa urządzeń srk z wykorzystaniem układów elektronicznych rozwijała się wraz ze zwiększeniem się dostępności standardowych cyfrowych układów scalonych o coraz większym stopniu integracji.

Pojawienie się przemysłowych sterowników mikroprocesorowych umożliwiło zastosowanie rozwiązań programowych do realizacji algorytmów działania urządzeń srk. We współczesnych urządzeniach srk algorytmy sterowania, przetwarzania i przechowywania danych są realizowane głównie w sposób programowy, zwykle w układach mikroprocesorowych, w których realizacja zadanego algorytmu odbywa się zgodnie z przechowywanym w pamięci programem. W pierwszych mikroprocesorach program był realizowany w jednej pętli programowej, a czas pojedynczego cyklu wyznaczała liczba linii kodu zawartego w programie. Takie rozwiązanie nie zapewniało wymaganego w urządzeniach srk poziomu bezpieczeństwa, co wymusiło stosowanie mechanizmów zwielokrotniania (redundancji sprzętowej) i porównywania wyników przetwarzania funkcji srk.

Następnie przez blisko 20 lat systemy srk, wykorzystujące sterowniki mikroprocesorowe, były modyfikowane wraz z rozwojem samych mikroprocesorów oraz powstawaniem coraz lepszych wersji oprogramowania, eliminujących błędy w warstwie programowej. Równocześnie w systemach tych dodawano coraz bardziej złożone moduły wyszukiwania błędów w realizowanych funkcjach sterowania, jak również moduły wyszukiwania uszkodzeń w urządzeniach realizujących te funkcje.

Zaletą programowej realizacji algorytmów działania urządzeń sterujących jest uniwersalność struktury oraz łatwość modyfikacji realizowanego algorytmu, sprowadzająca się do zmiany programu. Wadą rozwiązań programowych jest sekwencyjność ich działania powodująca, że cykl programu sterowania zależy liniowo od liczby instrukcji w tym programie. W jednoprocessorowych sterownikach przemysłowych nie ma możliwości współbieżnej obsługi

układów zewnętrznych (obsługa urządzeń zewnętrznych odbywa się metodą przerw), przy czym sterowniki te składają się zwykle z kilku układów scalonych.

Realizacja programowa, pomimo uniwersalnej struktury urządzenia, wymaga opracowania oprogramowania użytkowego, co wiąże się z dużą pracochłonnością i znacznie zwiększa koszt projektowanego urządzenia. Dodatkowym problemem mogą być trudności zapewnienia obsługi w czasie rzeczywistym wielu, pracujących współbieżnie, urządzeń zewnętrznych, co występuje w systemach sterowania ruchem w transporcie podczas obsługi urządzeń kontrolno - pomiarowych.

Obecnie zarysował się wyraźnie trend poszukiwania nowych rozwiązań, umożliwiających zwiększenie liczby realizowanych funkcji bez zmiany szybkości działania układu, co w rozwiązaniach programowych przekłada się na niezmiennosc liczby linii kodu realizowanych przez program. W związku z realizacją projektów Europejskiego Systemu Zarządzania Ruchem Kolejowym ERTMS (*European Rail Traffic Management System*) i znacznym zwiększeniem liczby funkcji realizowanych przez Europejski System Kontroli Pociągu ETCS (*European Train Control System*), konsorcjum firm *Invensys, Alstom, Bombardier, Siemens, Alcatel, Ansaldo Signal* poszukuje nowych rozwiązań, w których można by zastąpić technikę mikroprocesorową tak, aby było możliwe zwiększanie funkcjonalności urządzeń i spełnienie wszystkich wymagań dotyczących ERTMS.

Alternatywą dla rozwiązań mikroprocesorowych może być powrót do rozwiązań sprzętowych (elektronicznych), bądź sprzętowo-programowych, uwzględniających rozwój technologii standardowych i specjalizowanych układów scalonych.

Niezależnie od sposobu realizacji algorytmów sterowania, współczesne urządzenia i systemy srk muszą spełniać odpowiednie normy bezpieczeństwa. Dla nowych systemów srk muszą być spełnione wymagania ujęte w następujących normach:

- 1) PN-EN50126:2002: Zastosowania kolejowe – Specyfikacja niezawodności, dostępności, podatności utrzymaniowej i bezpieczeństwa;
- 2) PN-EN50128:2002: Zastosowania kolejowe – Łączność, sygnalizacja i systemy sterowania – Programy dla kolejowych systemów sterowania i zabezpieczeń;
- 3) PN-EN50129:2003: Zastosowania kolejowe – Łączność, sygnalizacja i systemy sterowania – elektroniczne systemy sygnalizacji związane z bezpieczeństwem.

Normy te definiują większość wymagań w stosunku do rozwiązań sprzętowych, programowych i sprzętowo-programowych. I tak sprzęt stosowany w urządzeniach i systemach srk powinien spełniać wymagania norm PN-EN50126 i PN-EN50129, natomiast programy realizujące funkcje sterowania powinny być zgodne z wymaganiami normy PN-EN50128. Jednak – nawet w

odniesieniu do wymagań dotyczących oprogramowania, sformułowanych w normie PN-EN50128 – pojawiają się głosy stwierdzające, że w stosunku do najnowszych rozwiązań systemów srk, norma ta nie określa pełnych wymagań potrzebnych do zdefiniowania modelu danych oraz przygotowania i utrzymania danych aplikacyjnych [5]. Dlatego też jako normę regulującą przygotowanie danych stosuje się często normę RIA 23 zgodną z PN-EN50128, jednak rozszerzoną o tablice dotyczące metod używanych do przygotowania danych, wraz z odpowiednią rekomendacją – w celu zapewnienia żądanego poziomu bezpieczeństwa SIL (*Safety Integrity Level*).

W przytoczonych normach nie natrafiliśmy na wymagania dotyczące wspomaganego komputerowo procesu specyfikacji bądź wytwarzania układów scalonych, przeznaczonych do systemów srk. Nie jest więc jasne co zrobić i jaką normę zastosować w przypadku, gdy program tworzony na etapie specyfikacji ma służyć jedynie do wytworzenia sprzętu. Pytanie to jest coraz częściej zadawane w krajach Unii Europejskiej, a szczególnie w Wielkiej Brytanii [17]. Do tego zagadnienia powrócimy we wnioskach, podanych na zakończenie artykułu.

## **2. STANDARDOWE I SPECJALIZOWANE UKŁADY CYFROWE**

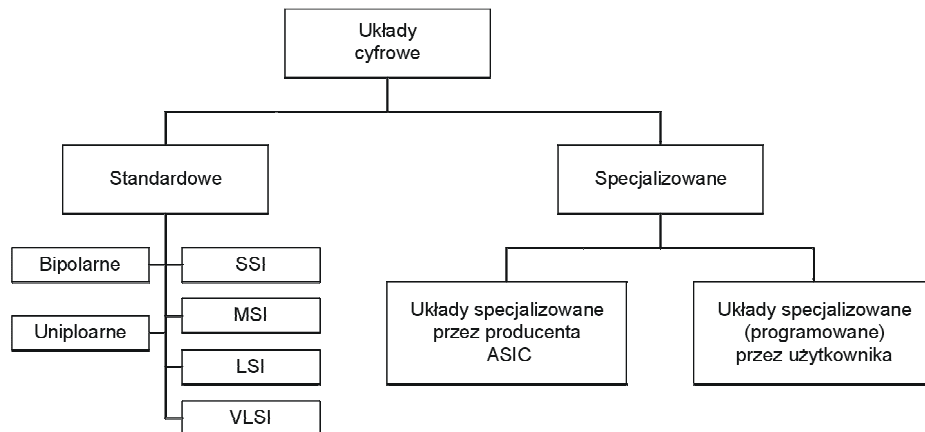
Sprzętowa (elektroniczna) realizacja algorytmów umożliwia bardzo szybką realizację algorytmu sterowania w sposób współbieżny, z wyeliminowaniem wad spowodowanych sekwencyjnością działania, występującą w programowej realizacji funkcji sterowania. Przy sprzętowej realizacji algorytmów sterowania, wartości sygnałów wejściowych określają sposób przetwarzania danych, natomiast równoległe rozprzestrzenianie się sygnałów w układach zapewnia naturalną współbieżność przetwarzania. Szybkość przetwarzania jest ograniczona wyłącznie czasem propagacji sygnałów w układzie.

Scalone układy cyfrowe mogą być podzielone na układy standardowe, zawierające typowe elementy lub bloki funkcjonalne (bramki, przerzutniki, rejestry, liczniki, pamięci itd.) ogólnego przeznaczenia, oraz na układy specjalizowane, realizujące ściśle określone przetwarzanie sygnałów (rys. 1).

Układy standardowe – ze względu na stopień scalenia – dzieli się na układy małej (SSI), średniej (MSI), oraz wielkiej (LSI, VLSI) skali integracji. Ze względu na zastosowaną technologię układy standardowe dzieli się na układy bipolarne oraz na układy unipolarne.

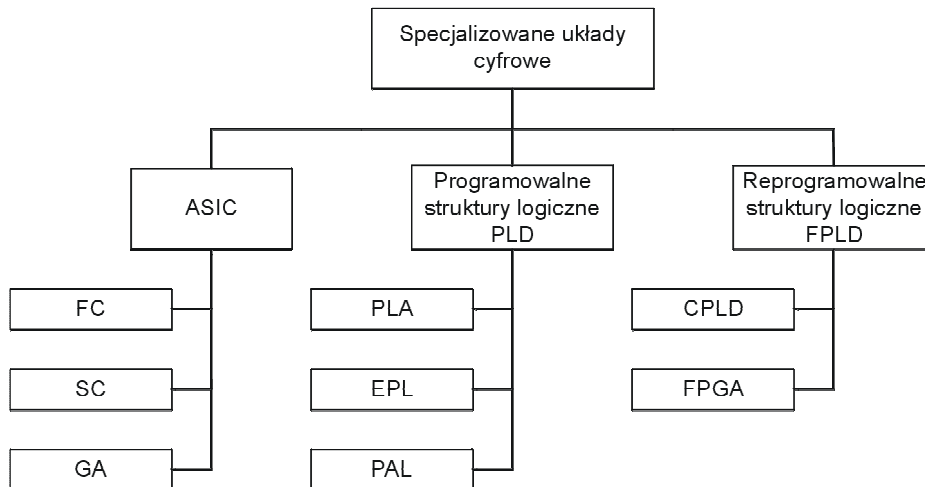
Rozwiązania sprzętowe urządzeń srk, z wykorzystaniem standardowych układów scalonych, prowadzą do rozbudowanej struktury urządzenia, zawierającego wiele różnorodnych układów

scalonych. Taka realizacja tych urządzeń może nie spełniać stawianych im funkcjonalnych, bezpieczeństwaowych i niezawodnościowych wymagań.



Rys. 1. Klasyfikacja układów cyfrowych

Specjalizowane układy cyfrowe można podzielić na układy specjalizowane przez producenta w procesie wytwarzania (ASIC) oraz na układy specjalizowane (programowane) przez użytkownika [12, 16].



Rys. 2. Klasyfikacja specjalizowanych układów cyfrowych

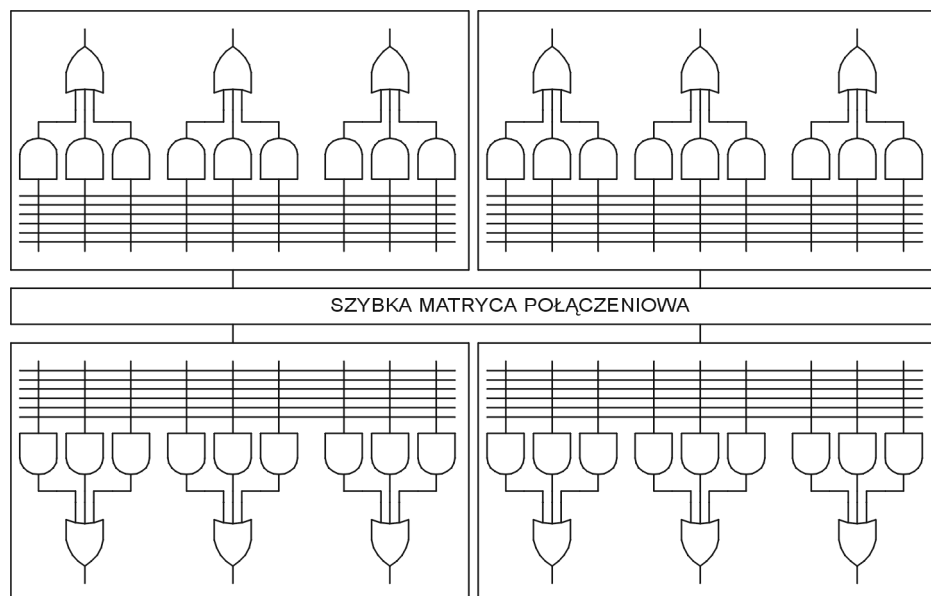
Specjalizowane układy ASIC (rys. 2), nazywane układami produkowanymi na zamówienie, mogą być wytwarzane: całkowicie indywidualnie z pełnym cyklem technologicznym FC (*Full Custom*), z wykorzystaniem komórek standardowych SC (*Standard Cell*) oraz z wykorzystaniem matryc bramek GA (*Gate Array*). Niezależnie od sposobu wytwarzania układów ASIC, cechuje je bardzo duży koszt wytwarzania. Dlatego też, mimo że układy te zapewniają możliwość uzyskania najlepszych parametrów czasowych i gęstości upakowania w krzemie, jest uzasadnione ekonomicznie stosowanie układów ASIC w produkcji wielkoseryjnej. Ponieważ

urządzenia srk wytwarzane są jednostkowo, bądź w krótkich seriach, zastosowanie w nich specjalizowanych układów ASIC nie ma obecnie uzasadnienia ekonomicznego.

Specjalizowane układy programowane przez użytkownika (rys. 2) można ogólnie podzielić na programowalne struktury logiczne PLD (*Programmable Logic Devices*) oraz na reprogramowalne struktury logiczne FPLD (*Field Programmable Logic Devices*).

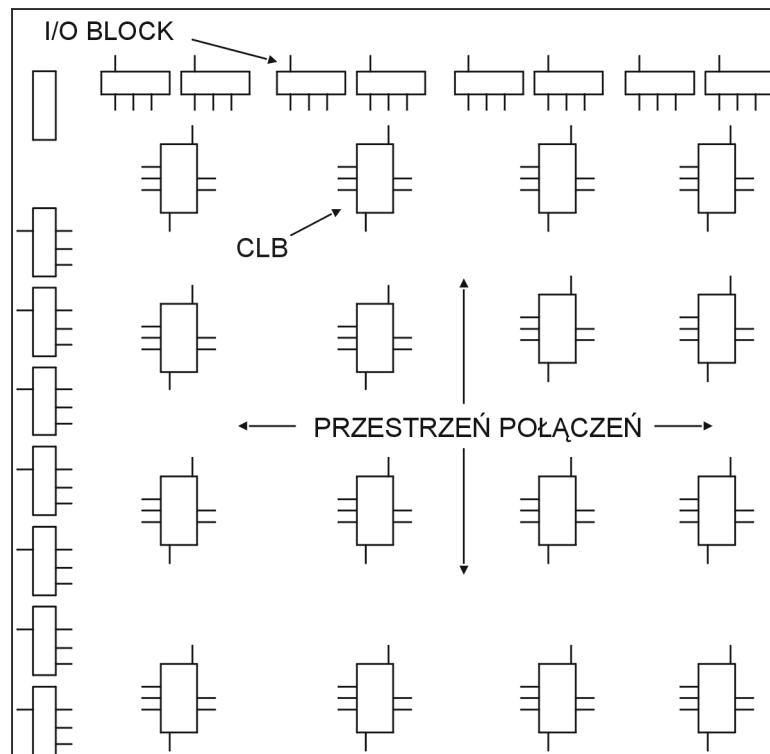
W programowalnych strukturach logicznych PLD do realizacji funkcji logicznych wykorzystywane są matryce AND oraz OR, przy czym programowalną może być jedna z matryc (matryca AND w układach PAL, matryca OR w układach EPL), albo obydwie matryce (w układach PLA). Wyposażenie układów PLD w przerzutniki synchroniczne umożliwia realizację w nich sekwencyjnych układów sterujących. Najbardziej popularne z układów PLD – układy PAL – są wytwarzane zarówno w wersji bipolarnej, jak i w wersji unipolarnej, z programowaniem jednokrotnym poprzez przepalenie zbędnych połączeń. Obecnie układy te są wytwarzane również w wersji umożliwiającej przeprogramowanie elektryczne (np. układy GAL).

Układy PLD, ze względu na zwykle małą liczbę wyprowadzeń, w urządzeniach srk mogą być wykorzystywane jako układy pomocnicze w sterownikach obiektowych. Przykładami mogą być: sterownik PLC MINICONTROL B&R, zawierający taki układ o małym stopniu scalenia GAL16V8B, bądź, stosowany na kolejach brytyjskich, zależnościowy system sterujący SSI (*Solid State Interlocking*), w którym układy PLD znalazły zastosowanie w sterownikach obiektowych. Należy jednak pamiętać, że w przedstawionych przykładach główną rolę odgrywają nadal procesory i napisane dla nich programy sterujące.



Rys. 3. Budowa układów CPLD

Reprogramowalne układy FPLD można podzielić na układy o budowie matrycowej CPLD (*Complex PLD*) oraz na układy o budowie komórkowej FPGA (*Field Programmable Gate Array*). Układy CPLD składają się w istocie z wielu układów PAL, połączonych szybko, programowaną matrycą połączeniową (rys. 3). Matryca ta zapewnia realizację połączenia bloków z wyprowadzeniami układu oraz realizację połączeń między blokami.



Rys. 4. Budowa układów FPGA

W układach FPGA (rys. 4) komórki zawierające konfigurowalne bloki logiczne CLB (*Configurable Logic Blok*) są otoczone globalnymi i lokalnymi liniami połączeń, oraz układami wejścia-wyjścia (I/O BLOCK). Komórki CLB zawierają: konfigurowalne tablice LUT (*Look-Up Table*), umożliwiające realizacje funkcji logicznych, przerzutniki synchroniczne oraz programowalne multipleksery.

Rozwój technologii układów scalonych i dostarczenie projektantom specjalizowanych układów cyfrowych w postaci programowalnych struktur logicznych FPLD umożliwiły w tworzonych sterownikach obiektowych i systemach sterowania połączenie elastyczności rozwiązań programowych z szybkością i współbieżnością działania, charakterystyczną dla rozwiązań sprzętowych. Dostępne obecnie rodziny układów FPLD, zawierające od kilku tysięcy do kilku milionów układów przeliczeniowych w jednej obudowie układu scalonego i mające od kilkudziesięciu do kilkuset wyprowadzeń typu wejścia-wyjścia, umożliwiają realizowanie w nich dowolnego algorytmu sterowania.

Algorytmy realizowane w wielu urządzeniach srk składają się głównie z funkcji sterujących oraz z prostych operacji przetwarzania i przechowywania danych, a więc szczególnie nadają się one do efektywnej realizacji sprzętowej [1]. Taka budowa układów sterowania zapewnia szybką realizację zadanych algorytmów sterowania, nie wymaga tworzenia oprogramowania narzędziowego, a dodatkowo zaimplementowanie tych algorytmów w jednym układzie scalonym umożliwia uzyskanie znacznie lepszych wskaźników niezawodnościowych w porównaniu z układami mikroprocesorowymi, realizującymi analogiczne algorytmy sterowania. Z tych powodów coraz częściej przemysłowe układy sterowania są realizowane w programowalnych strukturach logicznych. Projektowanie specjalizowanych układów sterowania w strukturach FPLD wymaga stosowania odpowiedniego wspomaganie komputerowego.

### **3. PAKIETY WSPOMAGANIA KOMPUTEROWEGO**

Systemy komputerowego projektowania układów sterowania powinny zawierać pakiety oprogramowania (narzędzia), wykorzystywane na etapach specyfikacji, weryfikacji, syntezy i implementacji projektowanego, specjalizowanego układu w programowalnej strukturze logicznej. Należy stwierdzić, że obecnie w przypadku współczesnych układów FPLD, brak jest wydajnych, zintegrowanych systemów komputerowego projektowania jednego producenta. Dlatego też należy, tworząc efektywny system komputerowego projektowania układów sterowania w strukturach FPLD, wybierać najlepsze narzędzia do poszczególnych etapów projektowania. Należy przy tym stosować na wszystkich etapach jednolitą platformę opisu, tak aby pliki wynikowe z jednego etapu mogły być wykorzystane na następnym etapie. Taką platformę stanowią języki opisu sprzętu HDL [7, 8, 12, 16].

Na etapie specyfikacji projektu, ze względu na uniwersalność narzędzi, należy wybierać te systemy komputerowego projektowania, które jako platformę opisu przyjmują standardowe języki HDL. Obecnie, zgodnie z IEEE, standardowymi językami opisu sprzętu są języki VHDL oraz *Verilog*. Język *Verilog* jest stosowany częściej do opisu układów elektronicznych, od poziomu krzemu – do prostych układów cyfrowych. Natomiast język VHDL jest bardziej przydatny do opisu złożonych układów i systemów sterowania, ze względu na możliwość opisu układów zarówno na poziomie funkcjonalnym, jak i strukturalnym oraz na łatwość tworzenia struktur hierarchicznych.

Specyfikacja układów sterowania o rozbudowanym algorytmie działania, bezpośrednio w języku VHDL nie jest łatwa, wymaga bowiem znajomości zasad tworzenia opisu w tym języku zarówno



jednostek projektowych, jak i zwłaszcza architektur układów. Dla projektantów, przyzwyczajonych do opisu projektowanych układów w postaci schematów ideowych, tworzenie specyfikacji układu bezpośrednio w kodzie źródłowym języka VHDL może być dużym utrudnieniem. Dlatego też, poszukując systemów komputerowego projektowania układów cyfrowych przydatnych do analizy i syntezy układów sterowania, należy rozpatrywać te systemy, które oprócz edytora tekstowego języków HDL, wyposażane są w edytory umożliwiające specyfikację układu w postaci powszechnie stosowanej przez projektantów układów sterowania. Z takiej postaci opisu system powinien umożliwiać automatyczne wygenerowanie kodu źródłowego języka HDL. Na przykład w systemie komputerowego projektowania *Active-HDL* taką funkcję pełnią: edytor grafów przejść FSM oraz edytor schematów blokowych BDE [8].

Ponieważ proces syntezy i implementacji algorytmów sterowania jest prowadzony dla konkretnych programowalnych struktur logicznych, więc dla zapewnienia optymalnej realizacji układu należy wybierać systemy, obsługujące jak największą liczbę rodzin i serii programowalnych struktur logicznych. Ponieważ najszerszą ofertę układów CPLD i FPGA proponują firmy *Altera* i *Xilinx*, rozpatrywano tylko te systemy, które zawierają biblioteki elementów tych firm. Z tych powodów do implementacji opracowanych algorytmów wybierano pakiety: dla *Altery* – Max +plus II oraz *Quartus*, a dla *Xilinx'a* – *Foundation* oraz *Alliance*. W celu zapewnienia efektywnego przebiegu procesu automatycznej syntezy system komputerowego projektowania musi być uzupełniony odpowiednimi pakietami, umożliwiającymi taką syntezę. Najlepiej gdy system zapewnia możliwość zintegrowania z nim odpowiednich pakietów syntezy i implementacji. Do najlepszych narzędzi syntezy można zaliczyć: *FPGA Express* firmy *Synopsys*, *Synplify* firmy *Synplicity*, *Leonardo Spectrum* firmy *Mentor Graphics*.

Z przedstawionych wyżej powodów, do specyfikacji i weryfikacji cyfrowych układów sterowania ruchem w transporcie zastosowano system *Active-HDL* firmy *Aldec* [8]. System ten umożliwia wybór domyślnego, standardowego języka opisu sprzętu (VHDL albo *Verilog*), efektywnych narzędzi syntezy oraz wybór rodzin i serii programowalnych struktur logicznych wielu producentów, wraz z odpowiednimi narzędziami implementacji (rys. 5).

Dodatkową zaletą systemu jest możliwość automatycznego testowania oraz porównywania (komparacji) przebiegów czasowych symulacji funkcjonalnej, z przebiegami uzyskanymi po syntezie, oraz po implementacji układu. Mechanizmy te ułatwiają w sposób istotny proces weryfikacji projektowanego układu.

## 4. PROJEKTOWANIE UKŁADÓW STEROWANIA W JEZYKU VHDL

Specyfikację kombinacyjnych układów sterowania w języku VHDL najprościej można przeprowadzić w edytorze tekstowym języka, z wykorzystaniem instrukcji współbieżnego przypisania sygnału [12, 16]. Ponieważ w układach tych stan wyjść jest jednoznacznie określony przez stan wejść, dla każdego z wyjść w kodzie źródłowym języka VHDL zostaje określona charakterystyka przypisania, z wykorzystaniem określonych w tym języku operatorów logicznych, addytywnych i multiplikatywnych.

Natomiast specyfikacja sekwencyjnych układów sterowania w języku VHDL jest bardzo żmudna i pracochłonna [7, 12]; znacznie efektywniejsze jest wykorzystanie edytorów grafów przejść automatów skończonych FSM oraz edytorów schematów blokowych BDE [7, 8].

Szczególnie przydatna jest możliwość specyfikacji układu w edytorze FSM w postaci grafu sterowania, bowiem opis sekwencyjnych układów synchronicznych w postaci grafów przejść automatów skończonych *Moore'a* albo *Mealy'ego* jest powszechnie stosowany w tradycyjnych metodach syntezy. Tak więc edytor FSM stanowi naturalne środowisko działania projektanta układów sterowania, a mechanizm specyfikacji i weryfikacji projektowanych układów sterowania w tym edytorze przedstawiono w pracach [7, 10].

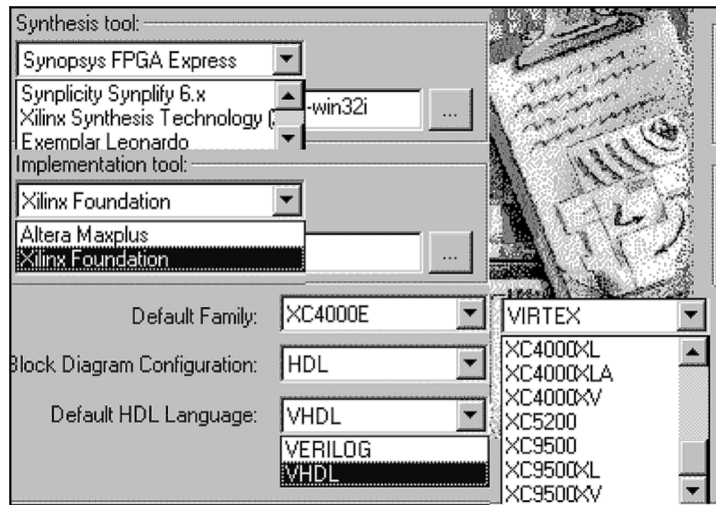
W edytorze FSM specyfikacja układu polega na utworzeniu w sposób graficzny jednostki projektowej oraz architektury układu. Jednostkę projektową (*entity*) tworzy się deklarując rodzaj, typ oraz liczbę portów projektowanego układu i w wyniku tych działań uzyskuje się graficzną postać opisu wyprowadzeń układu.

Architektura układu jest tworzona w postaci grafu przejść automatu, przy czym sygnały wyjściowe mogą być umieszczane zarówno na łukach grafu (tranzycjach), co jest charakterystyczne dla automatów *Mealy'ego*, jak i w wierzchołkach grafu, tak jak to występuje w przypadku automatów *Moore'a*.

Wyspecyfikowany w dowolnym edytorze projekt podlega kompilacji, w trakcie której wykrywane są błędy formalne opisu. Po ich usunięciu projektowany układ jest poddawany weryfikacji w trybie symulacji funkcjonalnej, w trakcie której jest sprawdzana poprawność logiczna działania projektowanego układu.

Sprawdzony w trybie symulacji funkcjonalnej algorytm działania układu jest poddawany za pomocą odpowiednich narzędzi (rys. 5) procesowi syntezy i implementacji w wybranej programowalnej strukturze logicznej określonej rodziny układów CPLD lub FPGA. Pomyślny przebieg tego procesu umożliwia przeprowadzenie symulacji czasowej, w trakcie której są

odwzorowywane opóźnienia czasowe, wnoszone przez poszczególne elementy wybranej programowalnej struktury logicznej.



Rys. 5. Wybór narzędzi syntezy i implementacji w pakiecie Active-HDL

Po funkcjonalnej i czasowej analizie poprawności działania projektowanych układów jest przeprowadzany proces prototypowania układów na specjalnych stanowiskach badawczych, wyposażonych w odpowiednie programowalne struktury logiczne. Autorzy przeprowadzali proces prototypowania na dwóch stanowiskach badawczych: stanowisku *workbench* VW 300, wyposażonym w programowalną strukturę *Virtex* XCV300 firmy *Xilinx*, oraz na stanowisku HES 800, zawierającym układ *Virtex* 800, umieszczony na specjalizowanej karcie podłączonej przez złącze PCI bezpośrednio do komputera PC (rys. 6).



Rys. 6. Widok specjalizowanej karty HES 800 stosowanej do prototypowania

Utworzone prototypy układów są wszechstronnie testowane; w trakcie testowania jest sprawdzana poprawność realizacji funkcji sterujących oraz są badane parametry czasowe działania układu.

## **5. OCENA WYBRANYCH PARAMETRÓW URZĄDZEŃ SRK ZREALIZOWANYCH W RÓŻNYCH TECHNOLOGIACH**

Przedstawione zalety programowalnych struktur logicznych i istniejących systemów komputerowego ich projektowania pozwalają sądzić, że ta technologia może być z sukcesem wdrażana w urządzeniach srk. Należy spodziewać się, że proces wdrażania struktur programowalnych nie będzie łatwy, podobnie jak dwadzieścia lat temu były problemy z akceptacją pierwszych zależnościowych systemów komputerowych. Jednak „przyjazność” procesów wspomaganą komputerowo specyfikacji i możliwość graficznej weryfikacji każdego etapu procesu projektowania wskazuje na to, że wdrożenie tej technologii może być znacznie prostsze i szybsze. Zastosowanie specjalizowanych układów cyfrowych nie zwolni jednak projektantów od tworzenia poprawnych algorytmów działania urządzeń srk.

Programowalne struktury logiczne będą stosowane w urządzeniach srk tylko wtedy, gdy spełnią wszystkie wymagania stawiane tym urządzeniom. Dlatego też dla przykładowego urządzenia srk – samoczynnej sygnalizacji przejazdowej (ssp) – zostaną oszacowane podstawowe parametry niezawodnościowe trzech sposobów jej realizacji:

- sprzętowej realizacji ze standardowych układów cyfrowych,
- programowej realizacji na sterownikach PLC,
- sprzętowej realizacji w programowalnych strukturach logicznych.

Pierwsze dwie realizacje są stosowane na kolejach w Polsce, a trzecie rozwiązanie z zastosowaniem układów FPLD zostało opracowane na Wydziale Transportu PW [9].

### **5.1. Warunki porównania**

Wszystkie urządzenia elektryczne, niezależnie od ich przeznaczenia, muszą spełniać wymagania dwóch ogólnych Dyrektyw UE: 73/23/EEC – dotyczącej niskonapięciowych urządzeń elektrycznych (oznaczanej skrótowo LVD) oraz 89/336/EEC – dotyczącej kompatybilności elektromagnetycznej (oznaczanej EMC). Elektroniczne urządzenia sterowania ruchem kolejowym powinny dodatkowo spełniać wymagania norm Unii Europejskiej, przedstawione w rozdziale 1: PN-EN50126:2002, PN-EN50128:2002 i PN-EN50129:2003. W tych normach są określone zagadnienia związane z niezawodnością, bezpieczeństwem, utrzymaniem oraz oprogramowaniem komputerowym, stosowanym w urządzeniach sterowania ruchem kolejowym. Określone są również zagadnienia dotyczące systemów elektronicznej sygnalizacji związanej z bezpieczeństwem. Producent urządzeń srk musi mieć Certyfikat Jakości ISO 9001. Po

przystąpieniu Polski do Unii Europejskiej urządzenia srk stosowane na kolejach w Polsce muszą spełniać wymagania dotyczące bezpieczeństwa systemów określone przepisami resortowymi [15] oraz zaleceniami Centrum Naukowo-Technicznego Kolejnictwa w Warszawie (CNTK) [19]. W tych opracowaniach obszernie omówiono zasady tworzenia oprogramowania, natomiast mniej uwagi poświęcono niezawodności rozwiązań sprzętowych.

W celu wyznaczenia takich wskaźników jak: intensywność uszkodzeń czy szybkość realizacji funkcji sterujących, dla porównywanych realizacji ssp przyjęto, że:

- porównywane są układy realizujące funkcje sterujące,
- porównywane będą pojedyncze kanały sterowania ssp dla typowego rozwiązania zastosowanego na linii dwutorowej,
- w sensie niezawodnościowym przyjęto dla elementów kanału szeregową strukturę elementów sprzętu oraz oprogramowania,
- dla wszystkich realizacji występują jednakowe warunki środowiskowe,
- wszystkie układy scalone zostały wyprodukowane według takich samych norm produkcyjnych.

Tak sformułowane założenia zapewniają identyczne warunki pracy oraz pozwalają przyjąć, że blok logiki ssp, niezależnie od zastosowanego sposobu realizacji, ma jednakową liczbę wejść i wyjść oraz realizuje te same funkcje sterujące.

## 5.2. Ocena intensywności uszkodzeń

W eksploatacji urządzeń bardzo ważne jest określenie zdolności urządzenia do spełniania postawionych mu wymagań w określonych dla niego warunkach i w danym przedziale czasu; zdolność ta jest nazywana niezawodnością  $R$  [6]. Najczęściej w technice do określenia niezawodności jest stosowany rozkład wykładniczy. Wskaźnikiem charakteryzującym niezawodność jest intensywność uszkodzeń  $\lambda$ , opisująca funkcję niezawodności według następującej zależności:

$$R(t) = e^{-\int_0^t \lambda(t) dt} \quad (1)$$

gdzie:

$R(t)$  – funkcja niezawodności,

$\lambda(t)$  – intensywność uszkodzeń.

Intensywność uszkodzeń określa prawdopodobieństwo wystąpienia uszkodzenia w wybranym przedziale czasu, w przyjętych warunkach eksploatacyjnych. Wyznaczenie wartości

intensywności uszkodzeń, przy znajomości struktury urządzenia, umożliwia ocenę niezawodności jego pracy, przyszłe koszty napraw oraz liczbę elementów zamiennych. Dla urządzeń sterujących nawet częściowa utrata własności na skutek zewnętrznych parametrów, wynikających z eksploatacji, może prowadzić do ich nieprawidłowej pracy, dlatego jest istotne określenie intensywności uszkodzeń parametrycznych  $\lambda_p$ . Jest ona w dużym stopniu miarą stabilności czasowej elementów podczas ich eksploatacji, zakładając uwzględnienie pełnego zestawu parametrów podlegających ocenie. Parametrami uwzględnianymi podczas wyznaczania intensywności uszkodzeń są: wielkość układu, rodzaj obudowy, liczba wyprowadzeń, rodzaj produkcji i stopień jej opanowania, temperatura i środowisko pracy, rodzaj układu (liniowy, cyfrowy). Obliczenia prognozowanej intensywności uszkodzeń parametrycznych zostały przeprowadzone na podstawie relacji i parametrów podanych w pracach [13, 14]. Zgodnie, z przyjętymi w punkcie 5.1. założeniami o jednakowych warunkach środowiskowych i produkcyjnych zostały określone dla rozpatrywanych realizacji wartości współczynników uwzględnianych w wyznaczaniu intensywności uszkodzeń (tabl. 1).

Tablica 1

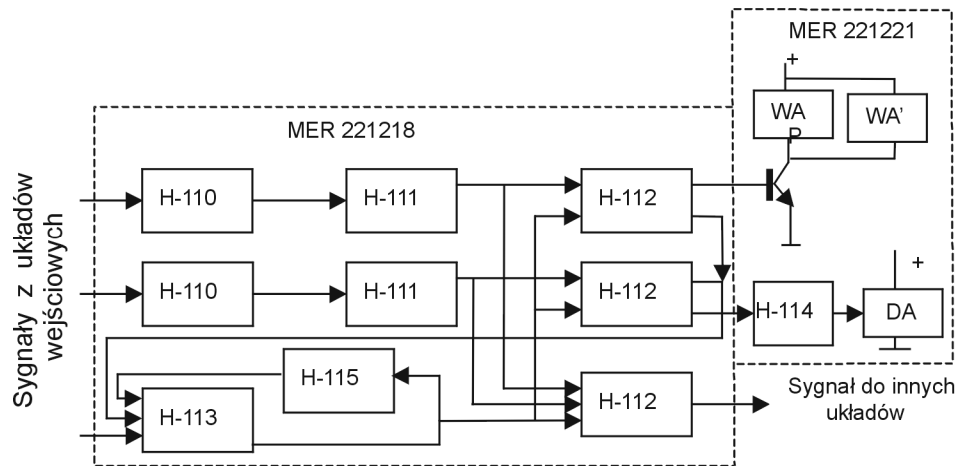
Współczynniki charakteryzujące warunki środowiskowe i sposób produkcji [16].

Symbol współczynnika	Wartość	Opis
$\pi_E$	2	Współczynnik charakteryzujący środowisko pracy. Dla ssp są to warunki środowiskowe $G_F$ – naziemne, stałe pomieszczenia bez klimatyzacji, z niewielkimi narażeniami na drgania mechaniczne
$\pi_Q$	1	Współczynnik charakteryzujący jakość produkcji w klasie B
$\pi_L$	1	Współczynnik charakteryzujący opanowanie produkcji, przyjęto że produkcja jest opanowana czyli trwa dłużej niż 2 lat.

### 5.2.1. Intensywność uszkodzeń dla ssp zbudowanej ze standardowych układów i elementów

Odpowiednio zaprojektowana konfiguracja połączeń układów zapewnia poprawną pracę układu logicznego. Na poprawną pracę takiego układu mają wpływ jedynie uszkodzenia wewnętrzne elementów scalonych oraz uszkodzenia połączeń między nimi. Przykładem rozwiązania wykorzystującego układy scalone jest ssp typu SPA-2A. Realizację funkcji sterujących w tym rozwiązaniu rozpoczynają dwa układy sterujące S1 oddzielnie dla toru 1 i 2. Układ ten w wersji MER221218 [2] zawiera 18 układów hybrydowych, 1 tranzystor, 12 diod, 11 kondensatorów, 35 rezystorów. Końcowy fragment funkcji sterujących jest realizowany przez układy H-114,

tranzystor, diodę oraz przekaźniki WA, WA', DA, znajdujące się na elemencie wykonawczym W (MER221221). Uproszczony przepływ sygnałów przedstawiony został na rysunku 7.



Rys. 7. Uproszczony schemat przesyłania sygnałów podczas realizacji funkcji

Zastosowane układy hybrydowe zawierają po kilka bramek NOR, rezystorów, diod i do trzydziestu połączeń wewnętrznych. Tylko układy H-114 oraz H-115 są układami liniowymi. Układy hybrydowe są wykonywane różnymi technologiami; część elementów (rezystory, kondensatory, połączenia wewnętrzne) są wykonane na płytce ceramicznej techniką cienkowarstwową, a inne elementy (diody, tranzystory, bramki) stanowią elementy dyskretne i są wmontowane w układzie. Na zewnątrz wyprowadzone są tylko punkty niezbędne do poprawnej pracy układu hybrydowego jako całości, ponieważ wykonane połączenia wewnętrzne określają konfigurację elementów, która nie może być już zmieniona, co ogranicza zastosowanie tych układów. Cały układ zalany jest masą żywiczną izolującą, która jest jednocześnie obudową. Zgodnie z przyjętymi założeniami, funkcje sterujące realizują układy hybrydowe, przekaźniki, diody i tranzystor, natomiast pozostałe elementy są związane z układami wejścia lub wyjścia. Na podstawie zależności (2) obliczono intensywności uszkodzeń poszczególnych typów układów hybrydowych.

$$\lambda_H = [N_c \lambda_c] (1 + 0,2\pi_E) \pi_F \pi_Q \pi_L \cdot 10^{-6} \quad [\text{h}^{-1}] \quad (2)$$

gdzie:

$\lambda_H$  – prognozowana intensywność uszkodzeń układu hybrydowego,

$N_c$  – liczba elementów typu  $c$  w układzie hybrydowym, (typ  $c$  oznacza np. diodę, tranzystor, bramkę)

$\lambda_c$  – intensywność uszkodzeń elementów typu  $c$ ,

$\pi_E$  – współczynnik wpływu środowiska,

$\pi_F$  – współczynnik funkcji realizowanych przez układ,

$\pi_Q$  – współczynnik jakości produkcji,

$\pi_L$  – współczynnik opanowania produkcji.

Zależność (3) opisuje intensywność uszkodzeń miniaturowych przekaźników WA, WA', DA, zastosowanych w realizacji funkcji sterujących:

$$\lambda_P = \lambda_b \pi_L \pi_C \pi_{CYC} \pi_E \pi_F \pi_Q \cdot 10^{-6} \quad [\text{h}^{-1}] \quad (3)$$

gdzie:

$\lambda_P$  – prognozowana intensywność uszkodzeń przekaźnika,

$\lambda_b$  – bazowa intensywność uszkodzeń,

$\pi_L$  – współczynnik charakteryzujący powierzchnię styku,

$\pi_C$  – współczynnik konfiguracji styków przekaźnika,

$\pi_{CYC}$  – współczynnik charakteryzujący liczbę wzbudzeń przekaźnika (na godzinę),

$\pi_E$  – współczynnik wpływu środowiska,

$\pi_F$  – współczynnik funkcji i konstrukcji przekaźnika,

$\pi_Q$  – współczynnik jakości produkcji, określanej jakością powierzchni styku.

Wyniki obliczeń intensywności uszkodzeń umieszczono w tabelicy 2. W obliczeniach tych pominięto wpływ uszkodzeń obudów i wyprowadzeń zewnętrznych oraz połączeń między poszczególnymi elementami na ogólną intensywność uszkodzeń układu.

Intensywność uszkodzeń układów realizujących funkcje sterujące ssp, zbudowanych z wykorzystaniem układów cyfrowych, opisuje zależność (4), bez uwzględnienia uszkodzeń połączeń między układami na płytkach drukowanych.

$$\lambda_{SPA-2A} = [2 \cdot (4\lambda_{H-110} + 4\lambda_{H-111} + 8\lambda_{H-112} + \lambda_{H-113} + \lambda_{H-114} + \lambda_{H-115}) + \lambda_D + \lambda_T + 3\lambda_P] \cdot 10^{-6} = 6,42 \cdot 10^{-6} \quad [\text{h}^{-1}] \quad (4)$$

gdzie:

$\lambda_{SPA-2A}$  – prognozowana intensywność uszkodzeń części sterującej, realizującej funkcje sterujące w ssp typu SPA-2A,

$\lambda_{H-1ij}$  – intensywność uszkodzeń układu hybrydowego H-1ij,

$\lambda_D$  – intensywność uszkodzeń diody,

$\lambda_T$  – intensywność uszkodzeń tranzystora,

$\lambda_P$  – intensywność uszkodzeń przekaźnika.



Prognozowana intensywności uszkodzeń elementów

Rodzaj elementów	Typ elementu	Liczba elementów	Typ MER-a	Intensywność uszkodzeń [ $10^{-6} \text{ h}^{-1}$ ]
Układy hybrydowe	H – 110	4	MER 221218/2	0,14
	H – 111	4	MER 221218/2	0,12
	H – 112	8	MER 221218/2	0,08
	H – 113	1	MER 221218/2	0,24
	H – 115	1	MER 221218/2	0,20
	H – 114	1	MER 221221	0,23
Elementy elektroniczne	dioda	1	MER 221221	0,007
	tranzystor	1	MER 221221	0,002
Przełączniki	MTd-6	2	MER 221221	0,68
	RM-82P	1	MER 221221	0,68

### 5.2.2. Intensywność uszkodzeń dla mikroprocesorowej realizacji ssp

Funkcje sterujące ssp są realizowane poprzez cykliczne przetwarzanie w sprzęcie odpowiedniego oprogramowania. Z tego względu poprawność pracy urządzeń ssp zrealizowanych w tej technice zależy od prawidłowego przetwarzania programu oraz uszkodzeń sprzętu. Przeprowadzone obliczenia intensywności uszkodzeń dotyczą sprzętu, natomiast dla programu oszacowano intensywność uszkodzeń. Do realizacji mikroprocesorowego ssp wykorzystano typowe sterowniki PLC, stosowane w przemyśle do sterowania procesami technologicznymi. Zgodnie z przyjętymi założeniami analizie uszkodzeń sprzętu zostały poddane karty procesorów, albowiem w nich są realizowane procesy logiczne. Do porównań wybrane zostały dwa typy sterowników – sterownik MINICONTROL firmy *Bernecker&Reiner*, stosowany w sygnalizacji typu SPA-4, oraz sterownik serii 90-30 firmy GE FANUC, zastosowany w ssp typu RASP-4.

Sterownik MINICONTROL ma kartę procesora oznaczoną symbolem CP-31. Zawiera ona procesor *Motorolla* 6303 oraz pamięci programu RAM 16KB i ROM 16KB (EEPROM) [3]. Pozostałe układy scalone oraz elementy pasywne, znajdujące się na tej karcie zostały pominięte w obliczeniach.

### Intensywności uszkodzeń układów mikroprocesorowych

Typ karty procesora	Prognozowana intensywność uszkodzeń [10 <sup>-6</sup> h <sup>-1</sup> ]		
	μP	RAM	ROM/Flash
CP-31	0,09	0,05	0,09
CPU-350	0,08	0,03	0,02

W sterowniku serii 90-30 firmy GE FANUC jednostka centralna CPU 350 zawiera procesor 80386EX oraz pamięć RAM 32KB, z podtrzymaniem bateryjnym i pamięć Flash 19,9 KB do przechowywania oprogramowania systemowego (*firmware*) [18].

W obu kartach procesorów pominięto, zgodnie z przyjętymi założeniami, intensywność uszkodzeń koprocessorów, rejestrów, magistral oraz układów związanych z ich zarządzaniem. Obliczenia intensywności uszkodzeń procesorów wykonano na podstawie zależności (5), a dla pamięci obliczenia przeprowadzono na podstawie zależności (6). Wyniki obliczeń poszczególnych układów zestawiono w tabelicy 3.

$$\lambda_{\mu P} = (C_1 \pi_T + C_2 \pi_E) \pi_Q \pi_L \cdot 10^{-6} \text{ [h}^{-1}\text{]} \quad (5)$$

gdzie:

$\lambda_{\mu P}$  – prognozowana intensywność uszkodzeń układów mikroprocesorów,

$C_1$  – współczynnik odzwierciedlający liczbę elementów w układzie (bramki przeliczeniowe),

$C_2$  – współczynnik typu obudowy oraz liczby wyprowadzeń,

$\pi_E$  – współczynnik wpływu środowiska,

$\pi_T$  – współczynnik wpływu temperatury złącza ( $T_J$ ),

$\pi_Q$  – współczynnik jakości produkcji,

$\pi_L$  – współczynnik opanowania produkcji.

$$\lambda_{PAM} = (C_1 \pi_T + C_2 \pi_E + \lambda_{cyc}) \pi_Q \pi_L \cdot 10^{-6} \text{ [h}^{-1}\text{]} \quad (6)$$

gdzie:

$\lambda_{PAM}$  – prognozowana intensywność uszkodzeń pamięci statycznych i dynamicznych,

$\lambda_{cyc}$  – prognozowana intensywność uszkodzeń wynikających z cyklicznych zmian odczyt / zapis z / w pamięci,

$C_1$  – współczynnik odzwierciedlający liczbę elementów w układzie ,

$C_2$  – współczynnik typu obudowy oraz liczby wyprowadzeń.

Intensywność uszkodzeń sprzętowych kart procesorów realizujących funkcje sterujące ssp zbudowanych w technice mikroprocesorowej opisują zależności (7) i (8), bez uwzględnienia połączeń między procesorem a pamięciami oraz układami pomocniczymi.

$$\lambda_{S\_CP-31} = \lambda_{\mu P} + \lambda_{RAM} + \lambda_{ROM} = 0,23 \cdot 10^{-6} \quad [h^{-1}] \quad (7)$$

gdzie:

$\lambda_{S\_CP-31}$  – prognozowana intensywność uszkodzeń sprzętowych karty procesora CP-31 sterownika MINICONTROL,

$\lambda_{\mu P}$  – prognozowana intensywność uszkodzeń mikroprocesora,

$\lambda_{RAM}$  – prognozowana intensywność uszkodzeń pamięci RAM,

$\lambda_{ROM}$  – prognozowana intensywność uszkodzeń pamięci ROM.

$$\lambda_{S\_CPU-350} = \lambda_{\mu P} + \lambda_{RAM} + \lambda_{Flash} = 0,13 \cdot 10^{-6} \quad [h^{-1}] \quad (8)$$

gdzie:

$\lambda_{S\_CPU-350}$  – prognozowana intensywność uszkodzeń sprzętowych karty procesora CPU-350 sterownika serii 90-30,

$\lambda_{\mu P}$  – prognozowana intensywność uszkodzeń mikroprocesora,

$\lambda_{RAM}$  – prognozowana intensywność uszkodzeń pamięci RAM,

$\lambda_{Flash}$  – prognozowana intensywność uszkodzeń pamięci typu Flash.

Na podstawie opracowania [11] oszacowano intensywność uszkodzeń wynikających z wykonywania programu, przyjmując jeden błąd wykonania instrukcji programowej na 1000 instrukcji źródłowych, na każde 1000 realizacji programu. Programy sterujące kanałami ssp różnią się, w zależności od zespołu programistów je piszących oraz zastosowanego przez nich języka programowania. Z tego względu, na podstawie pojemności pamięci programu, przyjęto dla każdego z rozwiązań liczbę instrukcji kodu źródłowego. Pamięć EEPROM współpracująca z kartą procesora CP-31 ma pojemność 4700 instrukcji programu. Zwykle cały program sterujący jest dzielony na moduły realizujące wyodrębnione funkcje. I tak w programie ssp występują moduły programowe realizujące: funkcje sterowania ssp, funkcje diagnostyczne urządzeń,

funkcje komunikacji zewnętrznej pomiędzy sterownikami oraz komunikacji z urządzeniami zdalnej kontroli, funkcje odmierzania w sposób programowy niezbędnych czasów (*timer`y*).

Dlatego też przyjęto, że moduły programowe, związane z realizacją logiki ssp, w tym rozwiązaniu nie przekraczają 200 instrukcji. Dla rozwiązania z zastosowaniem procesora CPU-350 przyjęto 220 instrukcji kodu źródłowego; zwiększenie liczby instrukcji wynika z innego sposobu sterowania, realizowanego przez sterownik w ssp typu RASP-4 [18]. Intensywność uszkodzeń oprogramowania rozpatrywanych ssp wyznaczono z zależności (9) i (10).

$$\lambda_{P\_CP-31} = \frac{\sum_{i=1}^n l_i}{\sum_{i=1}^n t_i} = 4,80 \cdot 10^{-6} \quad [\text{h}^{-1}] \quad (9)$$

gdzie:

$\lambda_{P\_CP-31}$  – prognozowana intensywność uszkodzeń wykonywania programu realizującego funkcje sterujące przetwarzanego w karcie procesora CP-31 sterownika MINICONTROL,

$\sum_{i=1}^n l_i$  – liczba linii programowych realizujących funkcje sterujące,

$\sum_{i=1}^n t_i$  – łączny czas wykonywania linii programowych, realizujących funkcje sterujące.

$$\lambda_{P\_CPU-350} = \frac{\sum_{i=1}^n l_i}{\sum_{i=1}^n t_i} = 3,96 \cdot 10^{-6} \quad [\text{h}^{-1}] \quad (10)$$

gdzie:

$\lambda_{P\_CPU-350}$  – prognozowana intensywność uszkodzeń wykonywania programu realizującego funkcje sterujące przetwarzanego w karcie procesora CPU-350 sterownika serii 90-30,

$\sum_{i=1}^n l_i$  – liczba linii programowych realizujących funkcje sterujące,

$\sum_{i=1}^n t_i$  – łączny czas wykonywania linii programowych, realizujących funkcje sterujące.

Łączna intensywność uszkodzeń sprzętu i przetwarzania programu sterującego rozwiązań programowych – realizowanych w kartach CP-31 i CPU-350 – wynosi odpowiednio:

$$\lambda_{CP-31} = \lambda_{S\_CP-31} + \lambda_{P\_CP-31} = 5,03 \cdot 10^{-6} \quad [h^{-1}] \quad (11)$$

$$\lambda_{CPU-350} = \lambda_{S\_CPU-350} + \lambda_{P\_CPU-350} = 4,09 \cdot 10^{-6} \quad [h^{-1}] \quad (12)$$

### 5.2.3. Intensywność uszkodzeń dla realizacji ssp w programowalnych strukturach logicznych

Na Wydziale Transportu Politechniki Warszawskiej zaprojektowany został blok logiczny realizujący funkcje sterujące ssp [9]. Blok ten, obejmujący całość funkcji sterujących ssp, został zaimplementowany w programowalnej strukturze logicznej FPGA, firmy *Xilinx* XCV4005E TQ144, zawierającej 5000 elementów przeliczeniowych. Przeprowadzona synteza i implementacja w tym układzie bloku logicznego ssp umożliwia określenie parametrów czasowych układu oraz wykorzystanie jego zasobów (tabl. 4).

Tablica 4

**Wyniki syntezy i implementacji logiki ssp w układzie XCV4005E TQ144**

Wykorzystanie zasobów struktury oraz wyznaczone parametry czasowe	Zasoby		
	Ogółem	Wykorzystane	% wykorzystania
Konfigurowalne bloki logiczne CLB	196	102	52
Uniwersalne bloki logiczne LUT 3 wejściowe	196	29	14
Uniwersalne bloki logiczne LUT czterowejściowe	392	186	47
Wejścia globalnych sygnałów zegarowych	4	1	25
Wyprowadzenia typu wejście wyjście (IOB)	112	82	73
Maksymalna częstotliwość pracy	21,318 MHz		
Maksymalne opóźnienie na połączeniach wewnętrznych	16,356 ns		

Niepełne wykorzystanie zasobów struktury do realizacji funkcji sterujących ssp umożliwia zaimplementowanie w pozostałej części bloków diagnostycznych lub modułów kontrolnych, bez pogarszania parametrów niezawodnościowych zaprojektowanego układu. Intensywność

uszkodzeń układu XCV4005 została obliczona z zależności (13), na podstawie opracowania [13] i wynosi:

$$\lambda_{XCV4005} = (C_1\pi_T + C_2\pi_E)\pi_Q\pi_L \cdot 10^{-6} = (0,003 \cdot 0,16 + 0,06 \cdot 2) \cdot 1 \cdot 1 \cdot 10^{-6} = 0,12 \cdot 10^{-6} \text{ [h}^{-1}\text{]} \quad (13)$$

gdzie:

$\lambda_{XCV4005}$  – prognozowana intensywność uszkodzeń struktury programowalnej XCV4005,

$C_1$  – współczynnik odzwierciedlający liczbę bramek przeliczeniowych (5000),

$C_2$  – współczynnik typu obudowy ( $TQ$ ) oraz liczby wyprowadzeń (144),

$\pi_E$  – współczynnik wpływu środowiska,

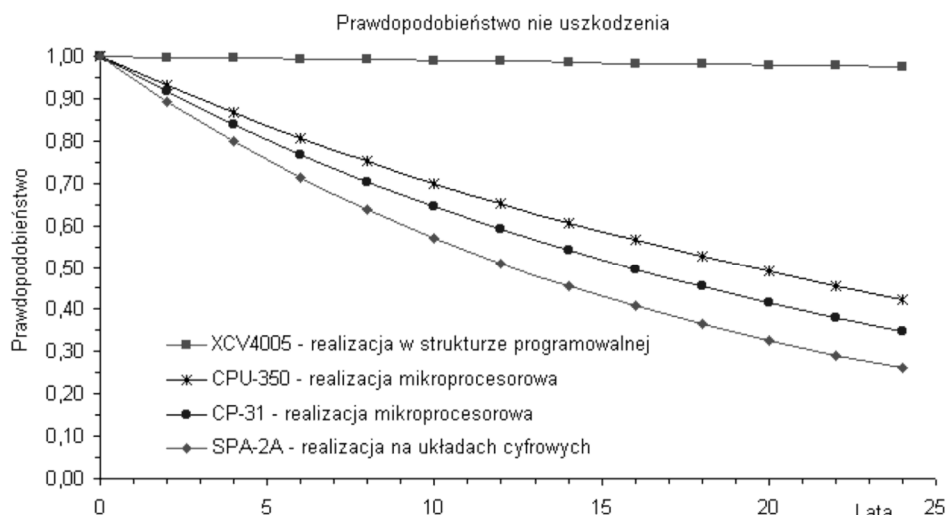
$\pi_T$  – współczynnik wpływu temperatury złącza ( $T_J$ ),

$\pi_Q$  – współczynnik jakości produkcji,

$\pi_L$  – współczynnik opanowania produkcji.

#### 5.2.4. Porównanie niezawodności analizowanych rozwiązań

Niezawodność określająca zdolność do poprawnej pracy urządzenia w przyjętych warunkach i czasie opisana zależnością (1), została przedstawiona na rysunku 8. Realizacja ssp typu SPA-2A, zbudowanej z cyfrowych i analogowych układów hybrydowych, elementów elektronicznych i przekaźników, charakteryzuje się dużą zawodnością, co przekłada się na dużą liczbę uszkodzeń. Bardziej niezawodne są mikroprocesorowe ssp, ale decydującym o ich niezawodnej pracy elementem jest poprawność przetwarzania programu sterującego. Niezawodność struktury programowalnej XCV4005 jest bardzo duża w dużym przedziale czasu, co dla układów sterowania zapewniających bezpieczeństwo ruchu jest bardzo ważnym argumentem, potwierdzającym przydatność układów programowalnych w nowo projektowanych urządzeniach srk.



Rys. 8. Prognoza prawdopodobieństwa poprawnej realizacji funkcji sterujących ssp

### 5.3. Ocena czasu realizacji funkcji sterowania

#### 5.3.1. Dla ssp zbudowanej ze standardowych układów i elementów

Sterowanie w sygnalizacji SPA-2A jest zrealizowane z wykorzystaniem układów hybrydowych, z których sygnał wyjściowy po wzmocnieniu steruje przekaźnikami WA, WA', DA (rys. 7). Najdłuższa ścieżka sygnału w tej ssp zawiera sześć bramek logicznych i przekaźnik. Dla standardowej serii układów cyfrowych czas propagacji dla pojedynczej bramki wynosi 10 ns. Natomiast zastosowane przekaźniki mają czasy wzbudzenia/odwzbudzenia odpowiednio: MTd-6 7,5/3 ms, a RM-82P 15/10 ms. Długie czasy zadziałania przekaźników umożliwiają pominięcie czasów propagacji sygnałów na połączeniach wewnętrznych w układach dyskretnych. Przyjęto więc, że czas realizacji funkcji sterowania w ssp typu SPA-2A jest zbliżony do najdłuższego czasu zadziałania przekaźnika (15 ms).

#### 5.3.2. Dla mikroprocesorowej realizacji ssp

Dla oprogramowanych sterowników PLC producenci określają całkowity czas przetwarzania zainstalowanego programu sterującego oraz podają czas przetwarzania 1000 linii programu sterującego. Programy sterujące realizujące funkcje sterujące ssp pracują w trybie cykliczno-sekwencyjnym; kolejno są wykonywane cykle programowe, zgodnie z przyjętym przez programistę algorytmem ich realizacji. W każdym cyklu jest przetwarzany moduł programowy, realizujący jedną funkcję sterującą, jej część lub jest realizowane zarządzanie oprogramowaniem. Następujące po sobie kolejne cykle programowe mogą zawierać różne

moduły programowe, co umożliwi w przyjętej liczbie cykli przetworzenie całego programu sterującego. Upraszczając można powiedzieć, że po sczytaniu stanu wejść związanych z realizowanym modułem przetwarzane są dane w celu określenia stanów wyjść sterowanych tym modułem lub określane są wartości zmiennych dla następnych modułów programowych realizowanych w kolejnych cyklach.

Procesor 6303 karty CP-31 (SPA-4) taktowany zegarem 1MHz przetwarza 1000 instrukcji w czasie 4 ms [3]. W zależności od liczby wejść oraz wyjść sterownika czas cyklu, w którym jest realizowany moduł programowy, wynosi od 15 do 25 ms. Czas przetwarzania całego programu jest zależny od liczby modułów i dla typowej sygnalizacji SPA-4 nie przekracza 150 ms. Przy przyjętej uprzednio liczbie instrukcji związanych z realizacją funkcji sterujących czas ich przetwarzania wyniesie 0,8 ms.

Procesor 80386EX zastosowany na karcie CPU350 sterownika sygnalizacji RASP-4 jest taktowany zegarem 25MHz. Umożliwia to wykonanie 1kB części logicznej programu w którym występują same styki w czasie 0,22 ms [4, 18]. Czas reakcji systemu od najechnania na czujnik włączający wynosi 500 ms dla tej ssp. Czas przetwarzania instrukcji związanych z realizacją sterownika tej ssp wyniesie 0,16 ms.

### **5.3.3. Dla realizacji ssp w programowalnych strukturach logicznych**

Dla zaimplementowanego rozwiązania realizującego funkcje sterujące ssp w układzie XCV4005E maksymalna częstotliwość taktowania układu wynosi 21,318 MHz. Okres zegara o tej częstotliwości odpowiada cyklowi pracy układu, przy którym zapewnione jest poprawne działanie układu. W procesie implementacji w układzie jest wyznaczane również maksymalne opóźnienie części kombinacyjnej układu, które wynosi 16,356 ns. Spełnienie łącznie tych dwóch warunków, zapewnia poprawną realizację procesu sterowania. Można zatem dla tego rozwiązania jako czas przetwarzania funkcji logicznych ssp przyjąć sumę minimalnego okresu zegara taktującego i opóźnień na połączeniach wewnętrznych. Dla tego rozwiązania maksymalny czas przetwarzania nie przekracza 63,264 ns.

### **5.3.4. Porównanie szybkości działania analizowanych rozwiązań ssp**

Porównując szybkości działania bloków logicznych analizowanych rozwiązań ssp należy stwierdzić, że największą szybkość działania wykazuje realizacja bloku logicznego w programowalnych strukturach logicznych. Najmniejszą szybkość działania ma, ze względu na zastosowane przekaźniki, blok logiczny ssp SPA-2A. Rozwiązania mikroprocesorowe zajmują



pozycje pośrednie, ustępując jednak w szybkości działania o ok. trzy rzędy wielkości, realizacji w układach FPGA. Czasy realizacji funkcji ssp zestawiono w tablicy 5.

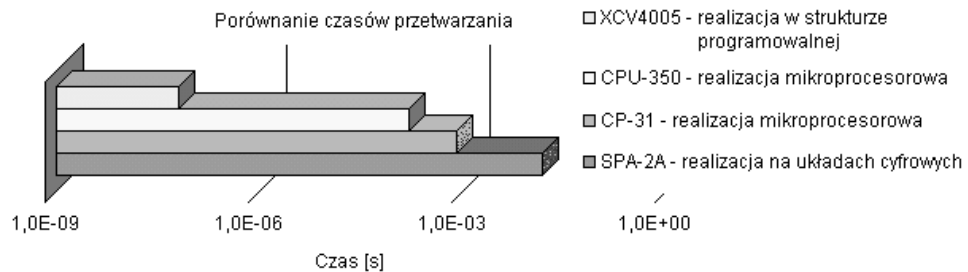
Tablica 5

**Zestawienie czasów realizacji funkcji sterujących.**

Sposób realizacji funkcji sterujących	Czas realizacji funkcji sterujących [s]
Układy cyfrowe (SPA-2A)	$15 \cdot 10^{-3}$
Sterownik Minicontrol (SPA-4)	$0,8 \cdot 10^{-3}$
Sterownik 90-30 (RASP-4)	$0,16 \cdot 10^{-3}$
Struktura programowalna (XCV4005)	$63 \cdot 10^{-9}$

Na rysunku 9 zobrazowano czasy przetwarzania funkcji sterujących z wykorzystaniem logarytmicznej skali czasu.

Mimo, że kryterium szybkości działania w prostych urządzeniach srk nie ma istotnego znaczenia, to jednak jego rola zwiększa się wraz ze zwiększeniem złożoności systemu sterowania (np. podczas przejścia do sterowania obszarowego). Dodatkowo duży zapas w szybkości działania umożliwia realizację algorytmów samotestowania w programowalnej strukturze logicznej.



Rys. 9. Porównanie czasu realizacji logiki ssp

**6. WNIOSKI**

Sprzętową realizację algorytmów działania urządzeń srk w programowalnych strukturach logicznych cechują duża szybkość działania oraz duże wskaźniki niezawodnościowe. Szeroka gama układów programowalnych i reprogramowalnych umożliwia wybór optymalnych struktur zarówno na etapie prototypowania, jak i w końcowej realizacji. Istniejące systemy komputerowego projektowania, wykorzystujące platformę języków opisu sprzętu umożliwiają specyfikację układów w różnych postaciach oraz automatyczną ich syntezę i implementację w

konkretnej strukturze logicznej. Przy czym poprawność tworzonych układów jest weryfikowana i testowana na wszystkich etapach procesu projektowania.

Problem bezpieczeństwa oprogramowania stosowanego do projektowania układów srk w programowalnych strukturach logicznych można porównać z bezpieczeństwem oprogramowania stosowanego do projektowania standardowych układów scalonych lub układów scalonych AISC. Zarówno w jednym, jak i w drugim przypadku oprogramowanie nie służy do realizacji funkcji sterowania, lecz do opisanie i wytworzenia sprzętu.

Ponieważ przedstawione we wstępie normy nie definiują wymagań odnośnie oprogramowania przeznaczonego do projektowania układów scalonych, należy przyjąć, że urządzenia srk – niezależnie od rodzaju zastosowanych układów scalonych (standardowych, specjalizowanych układów AISC czy wreszcie programowalnych struktur logicznych) – powinny być traktowane jako rozwiązanie sprzętowe. Jako takie powinny spełniać normy dotyczące sprzętu (tj. przedstawione w rozdziałach 1 i 5 normy PN-EN50126 i PN-EN50129). Takie podejście znajduje potwierdzenie w doświadczeniach komputerowego projektowania układów cyfrowych oraz w literaturze związanej z projektowaniem nowoczesnych systemów srk [17].

Przedstawione zalety realizacji algorytmów sterowania w układach FPLD pozwalają sądzić, że programowalne i reprogramowalne struktury logiczne będą coraz szerzej stosowane w sterownikach przemysłowych, w tym w urządzeniach srk.

## BIBLIOGRAFIA

1. *Dąbrowa-Bajon M.*: Podstawy sterowania ruchem kolejowym. Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 2002.
2. DTR-SPA-2A ABB ZWUS Signal Ltd, Katowice, 1992.
3. DTR-96/SPA-4 ABB ZWUS Signal Sp.zo.o. oraz Załącznik 1, Katowice, 1996.
4. DTR-2002/RASP-4F Z.A „KOMBUD” S.A. Radom, Warszawa, 2004.
5. *Faulkner A.*: Safe Data: The use of data in the context of a railway control system, in „Components of System Safety: Proceedings of the Tenth Safety-Critical System Symposium, Southampton 2002”.
6. *Głasysz H., Peciakowski E.*: Niezawodność elementów elektronicznych. WKŁ, Warszawa, 1984.
7. *Kawalec P.*: Analiz sistemi projektowania ustrojstw transportnoj awtomatiki w

- programmirujemych logiczieskich integralnych schiemiach. Międzynarodowe Międzyuczelniane Prace Naukowe „Aktualnyje problemy razvitiia tiehniczieskich sriedstw železnodorożnoj avtomatiki i tielemiechaniki”. Rostow, 2003, s. 130 – 135.
8. *Kawalec P.*: Zastosowanie układów PLD i FPGA w urządzeniach sterowania ruchem w transporcie. Materiały Międzynarodowej Konferencji Naukowo-Technicznej „Technika sterowania ruchem kolejowym u progu XXI wieku”, Warszawa, 1999, s. 96–100.
  9. *Kawalec P., Koliński D.*: Algorytmizacja funkcji samoczynnych sygnalizacji przejazdowych z wykorzystaniem wspomaganie komputerowego. Politechnika Radomska, Prace Naukowe – Transport, 2003 nr 1 (17), s. 255 – 260.
  10. *Kawalec P., Mocki J.*: Specification and verification of the interlocking functions for signaling devices using hardware description languages HDL. Proc. of the 11 International Symposium „Zel 2004” Railways on the Edge of the 3<sup>rd</sup> Millennium ”On the way towards the ‘European’ Railway- Harmonisation and IST”. Žilina, Slovakia, 2004, s. 41 – 46.
  11. *Lewiński A.*: Problemy oprogramowania bezpiecznych systemów komputerowych w zastosowaniach transportu kolejowego. Politechnika Radomska, *Monografie*, Radom, 2001 nr. 49.
  12. *Łuba T., Zbierzchowski B.*: Komputerowe projektowanie układów cyfrowych. WKŁ, Warszawa, 2000.
  13. Military handbook: electronic reliability design handbook, USA Department Of Defense, MIL-HDBK-338B (01.10.1998).
  14. Military handbook: reliability prediction of electronic equipment, USA Department Of Defense, raporty MIL 217F (02.12.1991), Notice 1 (10.07.1992), Notice 2 (28.02.1995).
  15. Norma zakładowa ZN-91/MtiGM-CBP-12, 1991. Bezpieczeństwo systemów sterowania ruchem kolejowym.
  16. *Pasierbiński J., Zbysiński P.*: Układy programowalne w praktyce. WKŁ, Warszawa, 2001.
  17. *Shannon I., Short R.*: Interlocking Development. IRSE News – Miesięcznik brytyjskiej Instytucji Inżynierów Sterowania Ruchem Kolejowym, 2005 nr 101.
  18. [www.astor.com.pl/gefanuc/katalog/9030/cpu350-360.html](http://www.astor.com.pl/gefanuc/katalog/9030/cpu350-360.html)
  19. Wymagania bezpieczeństwa dla urządzeń sterowania ruchem kolejowym, wymagania. Opracowane CNTK, Warszawa 1997.