Systemy modułowe

Jak widać, wiele układów eksperymentalnych używa tych samych, podstawowych bloków...
$\Rightarrow$ W laboratorium, gdzie często przebudowujemy układy, przydałoby się mieć zestawy pasujących do siebie i rozmontowywalnych „klocków”.

Także w stacjonarnych detektorach użycie masowo produkowanych modułów może obniżyć koszty.

Nawet jeżeli budujemy specjalną elektronikę, gotowe obudowy, zasilania i protokoły komunikacji ułatwiają pracę.

Warunek powodzenia takiego planu: **standaryzacja**:

- sygnałów,
- mechaniczna (pudełka, kable),
- zasilania.

Modularny standard elektroniki definiuje normalnie (jako minimum):

- wymiary modułów i standardowy ”stojak” (kasetę) na nie;
- system zasilania modułów w tej kasecie (napięcia, moc);
- konwencję przesyłania sygnałów cyfrowych (poziomy logiczne, obciążalność wyjść);
- standard kablowania (parametry kabli, wtyczki);

A często definiuje również

- protokół komunikacji cyfrowej między modułami za pośrednictwem złącza w samej kasecie.

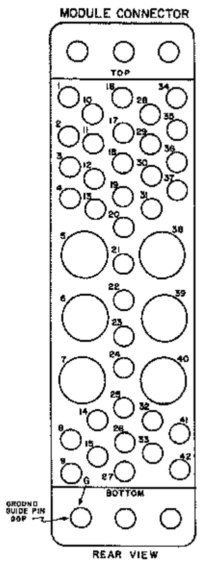
"The wonderful thing about standards is that there are so many of them to choose from" (unknown)

NIM

Nuclear Instrumentation Modules (Methods?)

Zaproponowany we wczesnych latach '60. Definiuje:

- wymiary zestandaryzowanego modułu: wys. 8.75″, szer. 1.35″. Moduł może mieć szerokość będącą wielokrotnością podstawowej.
- kasetę na moduły: mieści do 12 "pojedyńczych" modułów ustawionych obok siebie, daje się instalować w standardowym stojaku ("rack") szerokości 19″. Wyposażona w zasilacz i panel chłodzący (wentylatory).
- zasilania: kaseta dostarcza modułom zasilanie poprzez specjalne złącza na tylnej ściance. Dostarczane napięcia (co najmniej): ± 6 V, ± 24 V, często także ± 12 V i ~ 117 V.
  - kaseta nie zapewnia praktycznie żadnej łączności pomiędzy modułami (są zdefiniowane linie np. RESET czy GATE, ale nie są powszechnie używane)



- okablowanie: cienki kabel koncentryczny 50 Ω z wtyczkami LEMO.
- konwencję przesyłania sygnałów cyfrowych – a właściwie nawet dwie:

1. Slow NIM – powolne impulsy (czas narastania kilkaset ns), kodowanie poziomem napięcia (mierzonym na 1 kΩ): +4 do +12 V oznacza stan "1". Rzadko używany.
2. Fast NIM – czas narastania $\sim$1 ns, szerokość impulsu kilka ns. Kodowanie prądem mierzonym na impedancji 50 Ω: -14 do -18 mA oznacza logiczną "1". W przeliczeniu na napięcie jest to ok. -0.8 V.

Nadal powszechnie używany w laboratoriach, głównie na elektronikę analogową i prostą elektronikę cyfrową (bramki, koincydencje).

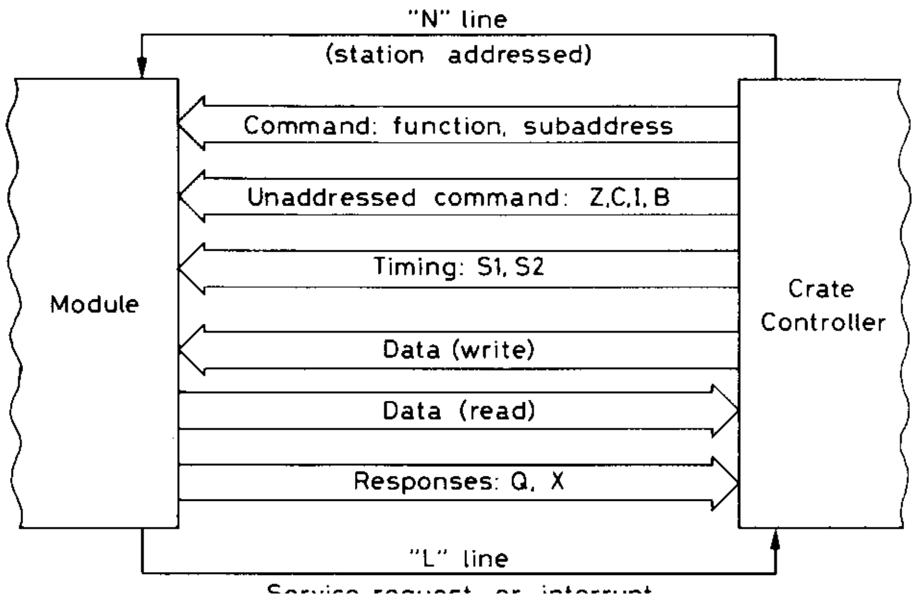
Standard CAMAC

Opracowany w koncu lat '60 i początku '70 w CERN, przyjęty jako standard ESONE (European Standards on Nuclear Electronics) potem także przez komitet NIM.

Idea – "uzupełnić" braki NIM w zakresie transmisji danych cyfrowych i budowy komputerowych systemów zbierania danych.

- Standardowy moduł – wysokość 221 mm (tyle co NIM), głębokość 305 mm, szerokość 17 mm (połowa NIM). Dopuszczalne moduły o szerokości wielokrotnej.
- Kaseta – wymiary podobne do NIM (nieco głębsza), mieści 25 stanowisk na moduły.
- Zasilania, chłodzenie – podobne jak NIM.
- Złącze na tylnej ściance – krawędziowe, 86 styków.
- Połączenia między kontaktami na tylnej ściance (dataway) umożliwiają przesyłanie danych cyfrowych. Poziomy sygnałów w standardzie TTL. CAMAC definiuje znaczenie sygnałów i protokół komunikacji.

Stanowisko 25 (najbardziej z prawej strony kasety) jest wyróżnione – w stanowiskach 24 i 25 musi znajdować się kontroler kasety. Tylko kontroler może inicjować transmisję danych. Transmisja możliwa jest tylko pomiędzy modułami a kontrolerem (nie pomiędzy samymi modułami).



Gniazda na tylniej ścianie kasety połączone są następującymi liniami:

- danych (R1–R24, W1–W24). Jak widać, możliwe jest przesłanie do 3 bajtów na cykl. Linie R służą do czytania danych (przesyłanie moduł wykonawczy → kontroler), linie W do ich pisania do modułu.
- wyboru modułu i funkcji: 24 indywidualne linie wyboru stanowiska (N), łączą każde stanowisko z gniazdem 25 (kontrolera). Do tego wspólne 4 linie wyboru podadresu (A) i 5 linii wyboru funkcji (F)
  - identyfikacją modułu jest jego pozycja w kasecie (numer stanowiska), kontroler chcąc skomunikować się z konkretnym modułem podaje impuls na jego linię N;
  - linie A mogą np. wybierać rejestr wewnątrz modułu;

  - linie F definiują rodzaj operacji. Niektóre z 32 możliwych kodów operacji zdefiniowane w standardzie (np. F0 – odczyt rejestru, F2 – odczyt z kasowaniem, F16 – zapis). Inne pozostawione do dyspozycji konstruktorów modułów.

- synchronizacji transmisji danych: B, S1, S2. Sygnały te generowane są w kontrolerze, moduły na ich podstawie i na podstawie sygnału N wykonują poszczególne fazy cyklu komunikacji. Względny timing tych sygnałów jest sztywno określony.

- wspólne sygnały kontrolne: clear (C), initialize (Z), inhibit (I).

  - sygnał C nakazuje wszystkim modułom skasować ich wewnętrzne rejestry
  - sygnał Z nakazuje zerowanie i inicjalizację modułu, głębszą niż sygnał C.
  - sygnał I nakazuje wszystkim modułom wstrzymanie ich normalnej działalności (np. przetworniki mają nie dokonywać konwersji, nawet jeżeli otrzymają bramkę). Stan ten trwa tak długo, jak długo kontroler utrzymuje stan "1" na linii I.

- odpowiedź modułu: X i Q

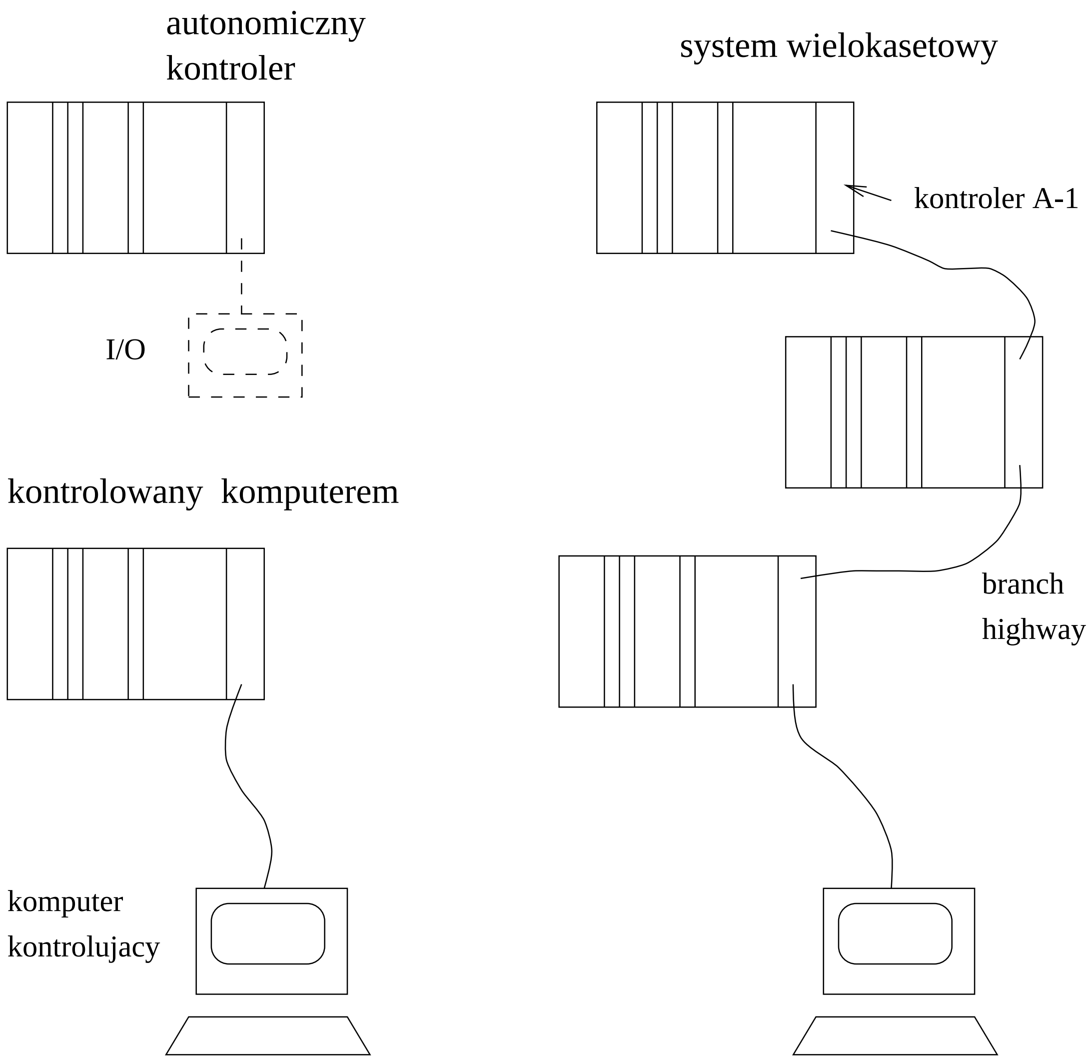
  - moduł wykonawczy musi zawsze wygenerować odpowiedź X gdy zostanie wywołany, inaczej kontroler uzna, że wystąpił błąd (np. stanowisko jest puste lub moduł jest uszkodzony.
  - Q - do uznania konstruktora modułu (np. koniec konwersji)

- indywidualne linie przerwań od modułów (Look-At-Me, L).

  - każdy moduł może zgłosić kontrolerowi w dowolnym momencie zajście jakiegoś zdarzenia, przez podanie poziomu "1" na linię L. Kontroler może, ale nie musi odpowiedzieć na takie "wezwanie".

Pojedyńczy cykl transmisji danych ma dokładnie określony przebieg czasowy i trwa 1 μs. Kontroler steruje poszczególnymi fazami przy pomocy sygnałów N, S1 i S2 (transmisja synchroniczna).

Możliwe kilka konfiguracji systemów opartych o CAMAC:



Systemy autonomiczne najczęściej oparte są o mikroprocesory, chociaż istnieją również dużo prostsze rozwiązania (kontrolery "ręczne" generujące pojedyńcze cykle lub krótkie sekwencje, programowane klawiszami).

Systemy wielokasetowe oparte są zazwyczaj o *branch highway* – kabel wieloparowy (84 pary skrętki), łączący do 7 kaset (CAMAC branch). Kasety muszą wtedy być wyposażone w zestandaryzowane kontrolery typu A–1. Komputer komunikuje się z modułami podając numer kasety, numer stanowiska w kasecie, podadres i funkcję modułu (CNAF).

Zalety i wady CAMAC:

- łatwość zestawiania układów eksperymentalnych odczytywanych i kontrolowanych programowo;
- łatwość budowy w warunkach laboratoryjnych niestandardowych modułów;
- akceptacja przez przemysł;
  - szeroko stosowany także poza laboratoriami fizyki (medycyna, sterowanie procesami technologicznymi)
- mała (jak na możliwości współczesnej technologii) prędkość transmisji, sztywne ograniczenia czasowe protokołu;
  - bywa, że CAMAC jest za szybki – np. jeżeli moduł kontroluje jakieś urządzenie położone daleko od kasety
- ciasny schemat adresowania (przeniesienie modułu w kasecie wymaga modyfikacji programu, 16 podadresów na moduł to za mało);
- trudności z budową dużych systemów.

Wyższa niż w NIM cena kasety powoduje, że oba systemy używane są często razem, elektronika nie wymagająca komunikacji cyfrowej (wzmacniacze, dyskryminatory, koincydencje etc) ustawiana jest w tańszych kasetach NIM.

FASTBUS

Opracowany w CERN w końcu lat siedemdziesiątych, planowany jako "następca" systemu CAMAC. Bardziej elastyczny, szybszy, z globalnym adresowaniem, z możliwością budowy bardzo dużych systemów. Niestety również trudny i skomplikowany w budowie i obsłudze.

Praktycznie nie wyszedł poza laboratoria fizyki jądrowej i cząstek elementarnych, a i w nich używany nieczęsto (tylko w dużych eksperymentach).

Standard VME

Powstał w podobnym czasie jak FASTBUS, zaproponowany przez firmę Motorola (podobieństwo z procesorami serii M68k nieprzypadkowe...). Przyjęty jako standard przez IEEE.

VME standaryzuje przede wszystkim komunikację cyfrową między modułami, inne cechy (np. wymiary kaset, modułów) drugorzędne.

Szyna (Bus) VME składa się z jednej lub dwóch specjalnych płyt *backplane* wyposażonych w 96-pinowe złącza w które włączane są moduły

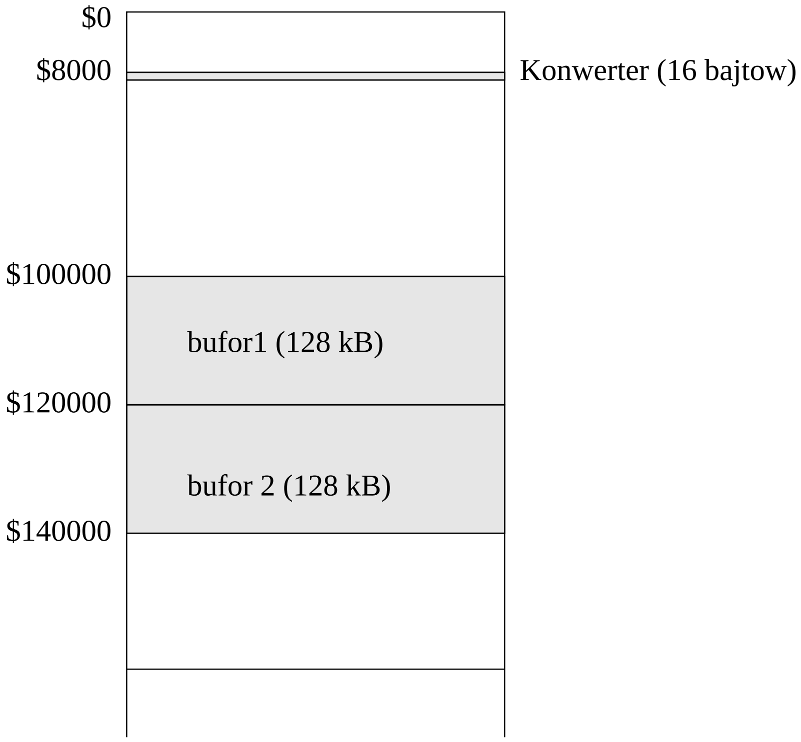
- Górny backplane (P1) łączy (prawie) wszystkie kontakty w złączach i przenosi wszystkie niezbędne sygnały kontrolne.
- Dolny backplane (P2) przesyła tylko dodatkowe bity adresu i danych – można używać VME w ograniczonej wersji bez niego.
- Tylko środkowy rząd kontaktów w złączach P2 używany przez VME, zewnętrzne mogą być wykorzystane przez użytkownika.

Komunikacja zachodzi zawsze pomiędzy parą modułów: jeden jest "Panem" (*Master*) – rozpoczyna transmisję, drugi jest "Sługą" (*Slave*): wykonuje zleconą operację i zakańcza cykl.

- Transmisja jest asynchroniczna – cykl trwa tak długo, jak długo master potrzebuje na przygotowanie operacji, a slave na jej wykonanie
- Duża elastyczność – możliwe zarówno bardzo szybkie przesyłanie danych, jak i dołączanie powolnych modułów (np. wolnych konwerterów albo łącz do odległych fizycznie urządzeń).

Wszystkie moduły na jednej szynie VME znajdują się we wspólnej "przestrzeni adresowej"

- 32 linie adresowe dają 32-bitowe adresy (na P1 tylko 24 bity). Dodatkowe 6 linii modyfikacji adresu (AM) pozwala efektywnie na rozszerzenie przestrzeni adresowej.
- Możliwe adresy 16-, 24- i 32-bitowe (w zależności od AM)
- Wystawiajac na szyny adresowe odpowiednią kombinacje adresu i AM master definiuje jednocześnie moduł z którym chce się komunikować i adres rejestru wewnątrz tego modułu.
  - "wywołany" moduł slave odpowie niezależnie od tego gdzie w kasecie znajduje się on i gdzie znajduje się master – łatwa przebudowa systemu
  - użytkownik musi zadbać o to, aby nigdy dwa moduły nie odpowiadały pod tym samym adresem VME. W praktyce moduły wykonawcze mają zwykle możliwość ustawienia (np. przełącznikami) "adresu bazowego" i, poczynając od niego, zajmują pewien obszar przestrzeni adresowej na swoje potrzeby



– możliwa bardzo duża liczba wewnętrznych podadresów w każdym module.

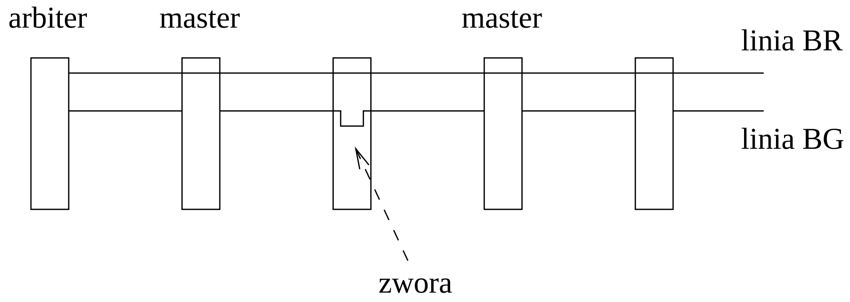
- Proponowane jest rozszerzenie VME do 64 bitów adresu.

Dane przesyłane są po 8, 16, 32 bity (tylko w systemach z szyną P2) lub 64 bity (nowy standard VME64 z multipleksowaniem danych na szynach adresowych). Master definiuje długość danych (ale uwaga! nie każdy moduł wykonawczy umie wykonać każdej długości transfer).

W jednej kasecie może być wiele modułów inicjujących przesyłanie danych (masterów). Moduł może być zarazem masterem i slave (tzn. dopuszczalne jest aby moduł zarówno inicjował transfery danych jak i odpowiadał na pewne adresy VME).

$\Rightarrow$ Konieczna synchronizacja dostępu do szyny:

- moduł najbardziej z lewej strony kasety (zwykle master, ale niekoniecznie) musi pełnić rolę nadzorcy (supervisor) i arbitra
- moduły proszą nadzorcę o dostęp do szyny przez wystawienie sygnału na linii Bus Request (jednej z czterech, priorytyzowane). Arbiter zezwala na dostęp przez odpowiedni sygnał Bus Grant:



Linie Bus Grant nie są połaczone wspólną szyną, tylko poprowadzone od złącza do złącza (styki BGIN i BGOUT).

– moduły slave po prostu przekazują sygnał dalej, zwierając BGIN z BGOUT;

– moduł master przekazuje sygnał dalej tylko jeżeli aktualnie nie żąda dostępu do szyny – w przeciwnym razie traktuje sygnał jako zezwolenie na dostęp i zatrzymuje go sobie;

⇒ moduł bardziej z lewej (bliżej nadzorcy) jest uprzywilejowany w sytuacji jednoczesnej próby dostępu;

⇒ w pustych stanowiskach trzeba instalować zworki między sygnałami BGIN i BGOUT żeby dalej położony master mógl funkcjonować.

– Nowoczesne złącza automatycznie zwierają odpowiednie linie jeżeli w stanowisku nie ma modułu.

- podobny jest mechanizm obsługi przerwań: moduł slave wystawia przerwanie na jednej z 7 linii IRQ. Master odpowiada sygnałem IACK, w analogiczny sposób przekazywanym pomiędzy modułami. Moduł utrzymujący przerwanie zatrzymuje IACK i podaje masterowi swą identyfikację;

- jeszcze jeden obowiązek supervisora – przerywanie niezakończonych cykli. Jeżeli przez określony czas (typowo około 50 μs) nikt nie zakończy rozpoczętego cyklu (żaden moduł "nie poczuwa się" do tego adresu), nadzorca wystawia sygnał *Bus Error* (BERR) i kończy cykl.

VME na ogół funkcjonuje jako system autonomiczny – w kasecie umieszczany jest moduł z komputerem, ewentualnie bus stanowi część komputera (np. starsze modele SGI). Istnieje ogromny wybór komputerów jednopłytowych – modułów VME które wymagają tylko dołączenia dysku i monitora aby stać sie pełnowartościowymi komputerami.

- przestrzeń adresowa VME jest w nich zazwyczaj widoczna jako część przestrzeni adresowej procesora – procesor widzi dane z innych modułów (konwerterów etc) jakby je od razu miał w swojej pamięci.

- w programie dostępy do VME wyglądają na ogól jak zwykłe instrukcje ładowania bądź podstawiania zmiennych (elementów tablic), tyle że zmienne te ulokowane są pod określonymi adresami.

VME ma ogromne poparcie w przemyśle (fizyka nie jest nawet bardzo liczącym się odbiorcą) – ogromny wybór modułów, rozwój technologii gwarantowany jeszcze przez jakiś czas.

W laboratoriach używany w budowie średnich i dużych systemów odczytu.

Transmisja danych cyfrowych

często trzeba przesłać odczytane już dane cyfrowe na odległości większe niż rozmiar kasety...

⇒ ... istnieją setki sposobów żeby to zrobić! Zależy jak daleko, jak szybko, między jakimi urządzeniami i co mamy do dyspozycji.

Kilka popularnych technik przesyłania danych:

RS-232 (czyli łącze szeregowe). Bradzo stary i bardzo popularny (praktycznie każdy komputer ma łącze szeregowe) – używany przede wszystkim do łączenia komputerów z terminalami i modemami.

- połączenie dwukierunkowe, w najprostszej wersji kablem trójżyłowym (Transmit, Receive, wspólne 0). Opcjonalnie sygnały potwierdzeń i kontroli – rzadziej używane;
- połączenie tylko *point-to-point*, czyli tylko pomiędzy dwoma urządzeniami;
- odległość transmisji do kilkudziesięciu metrów;
- szybkość ustawialna, ale zawsze niewielka (kilka kB/s to praktyczne maksimum). Mierzona w bitach na sekundę (baud);
- poza szybkością obie strony muszą się zgodzić co do innych parametrów (liczba bitów na bajt, parzystość...);
- wtyczki - najczęściej 25-pin "D-type". Uwaga: urządzenia RS-232 są w dwóch rodzajach: DCE i DTE, o zamienionych położeniach sygnałów TxD i RxD. łączenie tych samych rodzajów urządzeń (DCE z DCE lub DTE z DTE) wymaga kabli ze skrzyżowanymi sygnałami TxD i RxD.

Podstawowe zalety RS-232: powszechność i niewielkie wymagania sprzętowe (kabel może być praktycznie dowolnej jakości).

GPIB (znany też jako HP-IB albo IEC). Wymyślony przez firmę Hewlett-Packard, do zdalnego odczytu urządzeń pomiarowych.

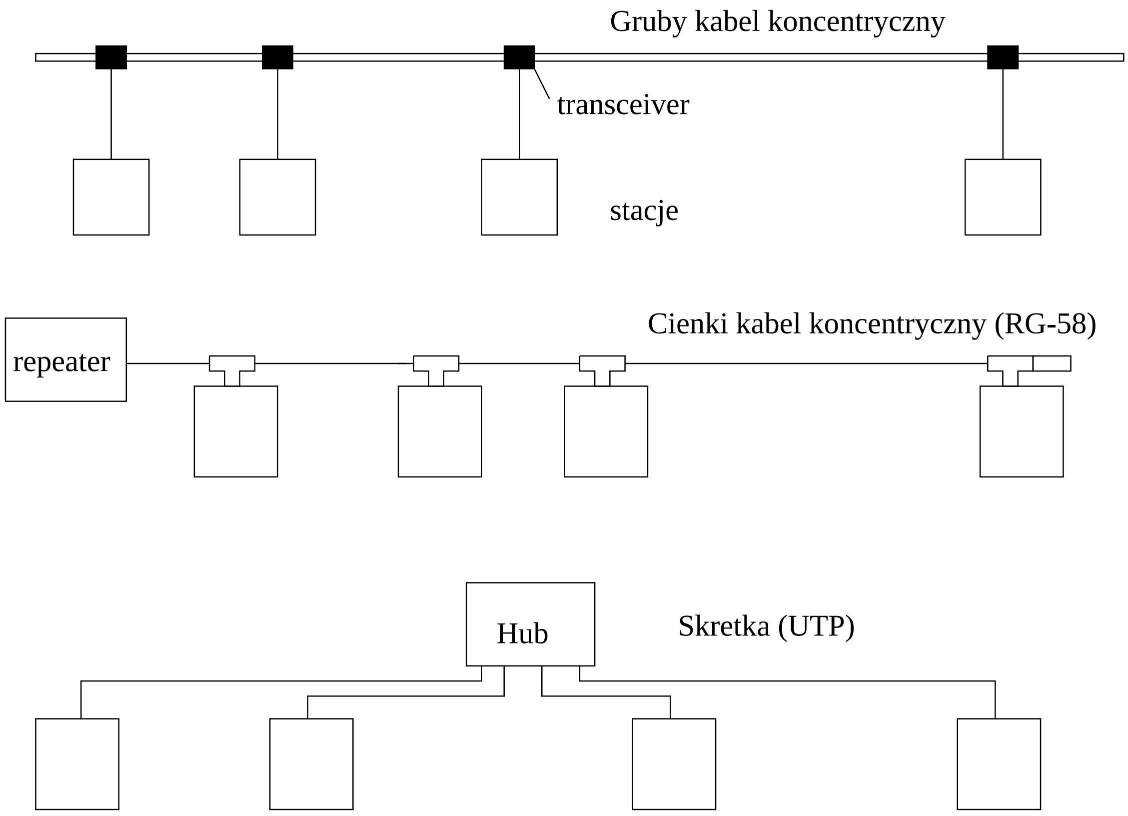
- Transmisja równoległa (16 sygnałów: 8 linii danych, 8 kontrolnych), asynchroniczna, dwukierunkowa;
- szybkość do kilkuset kB/s;
- możliwość podłączenia wielu urządzeń do jednego kontrolera (16 urządzeń, a z odpowienimi rozszerzeniami – nawet kilkaset);
- maksymalna długość kabla - kilkadziesiąt m (istnieją rozszerzenia zwiększające zasięg).

Użyteczne przede wszystkim przy odczytywaniu rozmaitych urządzeń kontrolnych (woltomierze, przepływomierze, czujniki temperatury etc) – bywają one fabrycznie wyposażone w odczyt GPIB. Wymaga dokupienia interface do komputera (np. modułu VME)

Ethernet: najczęstsze medium w lokalnych sieciach komputerowych. Większość komputerów ma wyjście Ethernetowe.

- transmisja szeregowa, po wspólnym kablu łączącym wszystkie komputery;
- szykość 10 Mbit/s (są szybsze wersje, ale nie są powszechnie używane);
- kilka standardów kablowania: gruby koncentryk, cienki koncentryk i skrętka (UTP);
- odległości do kilkuset metrów w zależności od rodzaju kabla (może być więcej, przy zastosowaniu odpowiednich połączeń pomiędzy segmentami).
- dane transmitowane są pakietami o długości max. 1536 bajtów, adresowanie pakietów 48-bitowym identyfikatorem stacji;

- synchronizacja dostępu do kabla – metodą "pierwszy-lepszy", z wykrywaniem kolizji.



Popularny i powszechnie dostępny, ale szybkość nie zawsze wystarczająca (i do tego nieprzewidywalne opóźnienia). Ze względu na skomplikowany protokół komunikacyjny umieją go tylko komputery (i wyspecjalizowany sprzęt sieciowy).

ATM (Asynchronous Transfer Mode) Technologia wymyślona przede wszystkim do budowy rozległych sieci telekomunikacyjnych (nie tylko transmisja danych!). Bazuje na koncepcji przełączania:

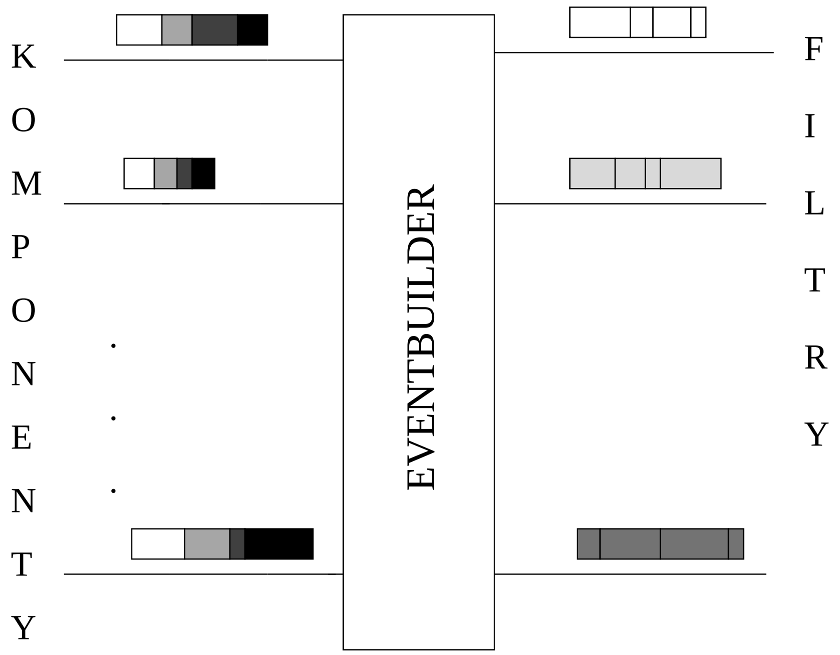
- dane przesyłane pakietami po 53 bajty, w tym 5 nagłówka (zawiera m.in. adres);

- połączenia szybkimi liniami (miedziane lub optyczne), tylko point-to-point. Szybkość transmisji najczęściej 155 Mbit/s lub 622 Mbit/s;

- podstawowy element sieci: przełącznik (switch): na bieżąco decyduje dokąd posłać otrzymywany właśnie pakiet – efektywnie dla każdego pakietu sieć zestawia połączenie miądzy nadajnikiem a odbiornikiem;
- uwaga: w razie konfliktów (dwa pakiety kierowane na tą samą drogę wyjściową) switch zgubi jeden z nich, nikomu o tym nie mówiąc! Protokół wyższego poziomu musi to wychwycić i zapewnić retransmisje w razie potrzeby.

Do czego to się nam może przydać? Np. do *Eventbuildera.*

W każdym dużym eksperymencie jest takie miejsce, gdzie dane z różnych części detektora "spotykają" się razem i są sklejane w kompletne przypadki, rozsyłane najczęściej do procesorów filtrujących:



Strumień danych przepływający przez eventbuilder może osiągać setki MB/s. Jak widać, trzeba zapewnić przy tym możliwość przesłania danych z każdego źródła do każdego odbiornika. Konstrukcja z przełączaniem i wieloma równoległymi przepływami danych jest pojemniejsza niż konstrukcja z jednym punktem przez który muszą przejść wszystkie dane („wąskim gardłem").

Oczywiście istnieje mnóstwo innych sposobów na przepchnięcie cyfrowych danych z komputera do komputera – czasem nawet budujemy własne rozwiązania, najlepiej pasujące do specyficznych potrzeb.

Elektronika w układach wyzwalania

Problem do rozwiązania: układ wyzwalania dużego eksperymentu wymaga wykonania dość złożonych obliczeń w bardzo krótkim czasie (pojedyńczych mikrosekund). Użycie mikroprocesorów jest więc wykluczone – nie zdążą nawet wczytać wszystkich danych!

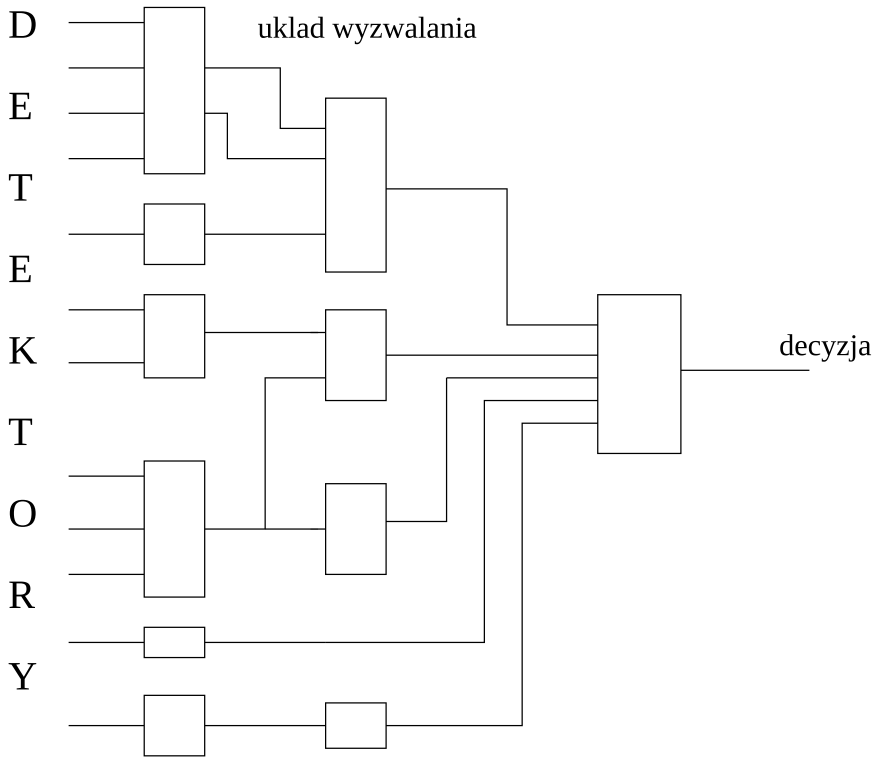
Przy akceleratorach o dużej częstości zderzeń dodatkowe utrudnienie: dane napływają z częstością repetycji akceleratora (nawet co kilkadziesiąt ns), układ wyzwalania musi więc jednocześnie rozpatrywać wiele przypadków.

Konieczne są:

1. odpowiednie algorytmy obliczeń: ustalony czas (liczba kroków), równoległość;
2. układy odpowiednio szybko realizujące te algorytmy.

Istotna jest też elastyczność algorytmu, czyli możliwość dostosowywania jego działania do zmiennych warunków eksperymentu (modyfikacja progów, zmiany kalibracji itp).

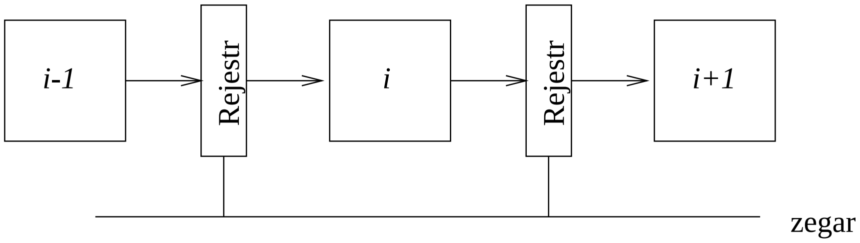
Zazwyczaj algorytmy systemów wyzwalania mają charakter "drzewowy" – kolejne kroki redukują ilość informacji, aż pozostaje jeden bit: decyzja.



Jeżeli zaaranżujemy powyższe drzewo tak, aby każda „gałąź” liczyła tyle samo kroków i każdy krok zajmował tyle samo czasu, dostaniemy algorytm zdatny do użycia w akceleratorze o wysokiej częstości przecięć.

⇒ Trzeba pamiętać o czasie przesyłania informacji (opóźnieniach w kablech), w dużym eksperymencie jest istotny.

Pomiędzy poszczególnymi krokami obliczeń ustawiane są zazwyczaj rejestry bramkowane wspólnym zegarem, które ”regulują” przepływ danych.



Ale co wykonuje te kolejne kroki obliczeń?

Na każdym kroku mamy do czynienia z operacjami logicznymi, przekształcającymi n bitów wejścia w k bitów wyjścia (czasami krok obliczeń jest automatem skończonym – wyjście zależy nie tylko od aktualnego stanu wyjść, ale także od poprzednich). Zgodnie z podstawowymi prawami logiki każde takie obliczenie można zrealizować za pomocą skończonej liczby elementarnych bramek logicznych.

⇒ Można teoretycznie! W praktyce liczba niezbędnych bramek może być bardzo duża (rozmiar układu, czas obliczeń). Do tego realizacja z bramek nie jest elastyczna: zmiana algorytmu wymaga zmiany połączeń pomiędzy nimi.

Dlatego często stosowana jest technika *lookup table*: zawczasu wyliczamy dla każdego możliwego stanu wejść pożądany stan wyjść i umieszczamy te dane w pamięci cyfrowej. Stan wejść traktujemy jak adres komórki pamięci, w której zapisany jest pożądany wynik.

Do dyspozycji mamy kilka technologii pamięci:

- RAM – czyli pamięć której zawartość można swobodnie zmieniać. Najbardziej elastyczne rozwiązanie, ale najdroższe, no i wymaga uzupełnienia systemu o mechanizm zapisu pamięci.

  ⇒ do naszych celów użyteczne są tylko pamięci statyczne (S-RAM), droższe i gabarytowo większe od używanych przez większość komputerów pamięci dynamicznych. Te ostatnie wymagają co kilka ms wykonania tzw. cyklu odświeżania, w czasie którego pamięć "przepisuje się" na nowo.

- PROM – czyli pamięć tylko do odczytu. Można ją zaprogramować na zewnątrz układu, przy pomocy specjalnego urządzenia. Późniejsze zmiany nie są jednak możliwe (można tylko kostkę pamięci wyjąć i wyrzucić, a na jej miejsce włożyć inną). Za to nie tracimy zawartości przy każdym wyłączeniu układu.

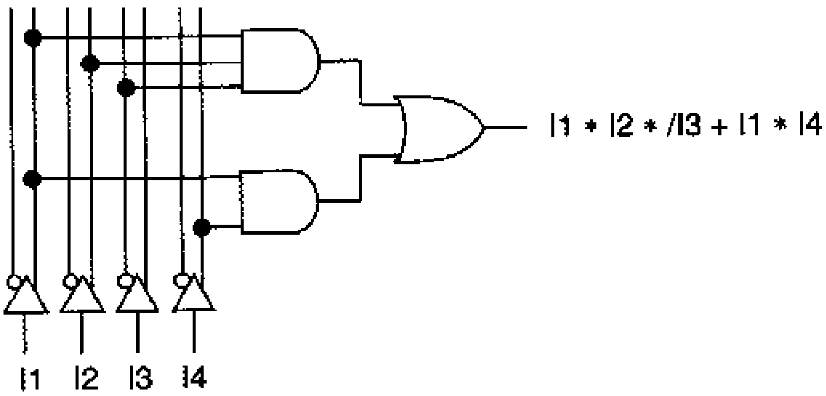
- EPROM – odmiana pamięci PROM, której w razie potrzeby przeprogramowania nie trzeba wyrzucać, można bowiem zawartość kostki skasować i zaprogramować pamięć na nowo. Kasowanie odbywa sie promieniowaniem UV lub elektrycznie.

Obszary zastosowania PROM i EPROM nie pokrywają się do końca – EPROM-y są zwykle większe (więcej bitów adresu i danych) niż PROM-y.

Wady techniki lookup table: rozmiary (i cena) pamięci rosną bardzo szybko z liczbą bitów na wejściu (każdy bit wejścia podwaja rozmiar pamięci). Z tegoż powodu trudno tak budować automaty skończone, czyli urządzenia o działaniu zależnym od obecnego i poprzednich stanów wejść.

O ile funkcja logiczna zamieniająca wejście na wyjście nie jest zbyt skomplikowana, można użyć prostszych programowalnych układów logicznych – **PAL**.

Zasada działania: każdą funkcję logiczną można przedstawić jako sumę iloczynów logicznych argumentów. PAL ma wewnątrz kilka bramek AND (realizujących składowe iloczyny) i macierz programowalnych połączeń, umożliwiającą podłączenie do wejść bramek każdego z sygnałów wejściowych (zaprzeczonego lub nie). Programowanie wygląda podobnie jak w wypadku pamięci PROM: w odpowiednim urządzeniu następuje ”przepalenie” wymaganych połączeń.



Układy PAL różnią się od siebie liczbą wejść, wyjść i liczbą wewnętrznych bramek AND.

A co z bardziej skomplikowanymi obliczeniami? Czasem chciałoby się mieć specjalnie dla siebie zrobiony układ scalony!

⇒ koszt zaprojektowania i zbudowania średnio skomplikowanego układu scalonego idzie w miliony dolarów – (prawie) zapomnieć!

Ale na szczęście producenci układów scalonych wychodzą nam troche naprzeciw, budując układy **Gate Array**

Projektowany jest układ scalony zawierający wiele elementarnych bramek logicznych (do kilkudziesięciu tysięcy), ale żadnych połączeń między nimi!

Użytkownik otrzymuje program pozwalający mu na opisanie funkcji żądanego układu (schematem lub opisowo) i planujący na tej podstawie brakujące połączenia. W pakiecie jest też symulator, pozwalający na sprawdzenie działania projektowanego układu przed jego realizacją.

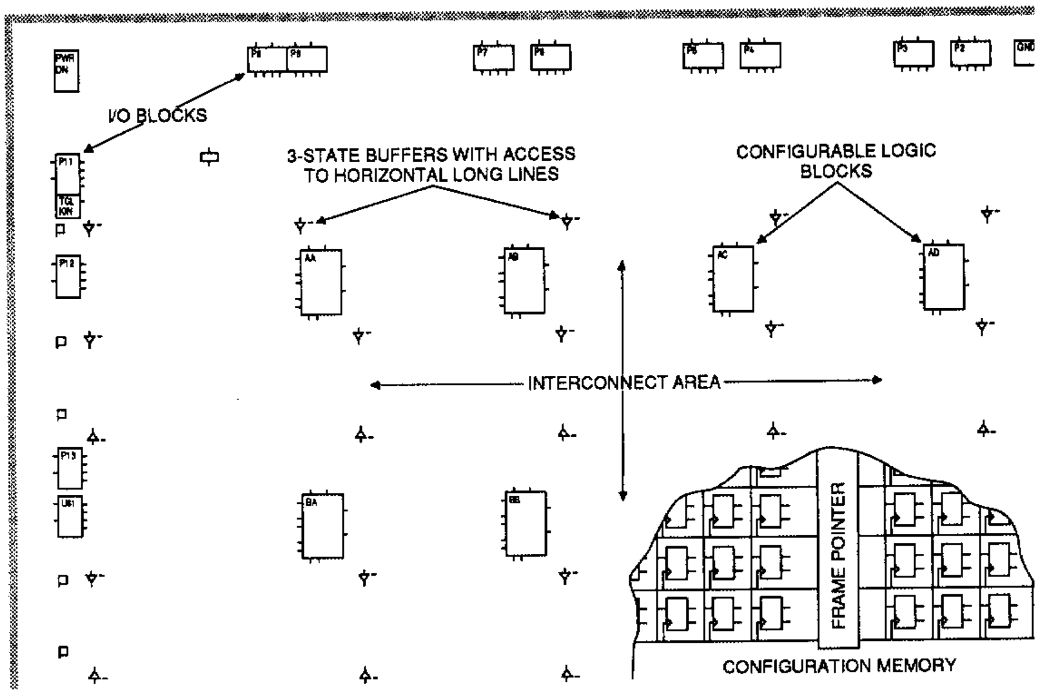
Na podstawie dostarczonego projektu producent przygotowuje brakujące maski z połączeniami i produkuje układy. Koszt: kilkadziesiąt k$ za przygotowanie masek, kilka $ za każdą kostkę.

- Opłacalne tylko jeżeli potrzebujemy większej liczby identycznych scalaków (tak od kilku tysięcy);
- trudno się modyfikuje w razie potrzeby.

Na szczęście istnieją wersje tej technologii programowalne "w domu": **Field Programmable Gate Arrays (FPGA)** (Xilinx, Altera).

Wewnątrz urządzenia znajdziemy pewną liczbę bloków logicznych (o architekturze PAL-i lub lookup table), rejestrów i sieć programowalnych połączeń pomiędzy tymi blokami.

Przykład: wnętrze układu Xilinx



Projektowanie układu jak przy zwykłych GA: opis funkcjonalny układu (graficzny lub tekstowy) jest "kompilowany" do postaci programu dla układu (spisu funkcji bloków logicznych i połączeń pomiędzy nimi). Program taki może być następnie wpisany do pamięci FPGA.

A pamięć ta może być:

- typu EPROM, czyli układ jest programowany w specjalnym urządzeniu, w razie potrzeby może być skasowany ultrafioletem lub elektrycznie i zaprogramowany inaczej;
- typu RAM – program musi być wpisywany do układu po każdym włączeniu. Źródłem może być zewnętrzna pamięć PROM, można też wpisywać program pod kontrolą komputera.

Zalety to możliwość upakowania skomplikowanych funkcji w niewielkiej objętości i możliwość modyfikowania funkcjonalności w razie potrzeby. Cena układu jest wyższa niż gate array-a (ale nie taka znowu wysoka), ale koszt projektowania jest wyraźnie niższy.

Idealne do stosowania nie tylko w układach wyzwalania, ale i w szybkich systemach odczytu.

Zamiast podsumowania

- nie ma współczesnego detektora fizyki wysokiej energii bez elektronicznego systemu odczytu...
- nie ma więc szans być fizykiem-eksperymentatorem i uniknąc jakiegokolwiek kontaktu z elektroniką...
- ...i chyba nie ma sensu unikać – bo jest to jak najbardziej do zrozumienia i może być bardzo ciekawe.