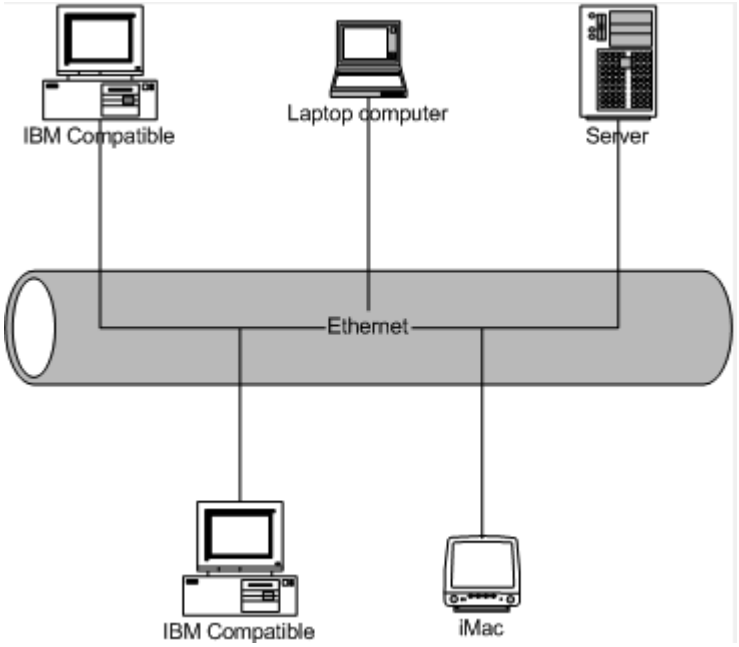
Wykład 11

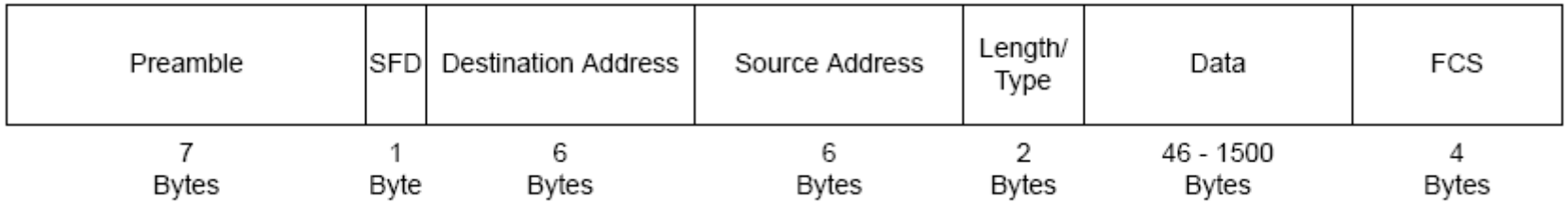
Obsługa portu Ethernet

Ethernet

- Ethernet to technologia, w której zawarte są standardy wykorzystywane w budowie głównie lokalnych sieci komputerowych. Jego specyfikacja została podana w standardzie 802.3 IEEE.
- Obejmuje ona specyfikację kabli oraz przesyłanych nimi sygnałów.
- Ethernet opisuje również format ramek i protokoły z dwóch najniższych warstw Modelu OSI.

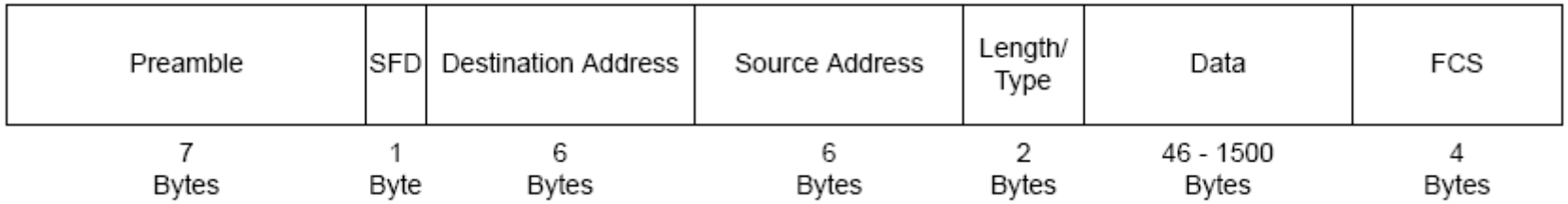


Ramka Ethernet



- Wszystkie pola ramki transmitowane są od strony lewej do prawej. Bity w poszczególnych polach transmitowane są startując od najmniej znaczącego bitu.
  - Preamble – preambuła wykorzystywana jest do synchronizacji czasów przesyłania ramki, składa się z 7 bajtów złożonych z naprzemiennych jedynek i zer
  - SFD (Start Frame Delimiter) – sekwencja bitów 1010.1011, oznaczająca początek ramki,
  - Destination Address – pole zawiera adres docelowy przesyłanej ramki, najmniej znaczący bit (LSB) określa czy jest to adres indywidualny (0), czy rozgłoszeniowy (1),
  - Source address (SA) – pole zawiera adres urządzenia wysyłającego ramkę,

Ramka Ethernet

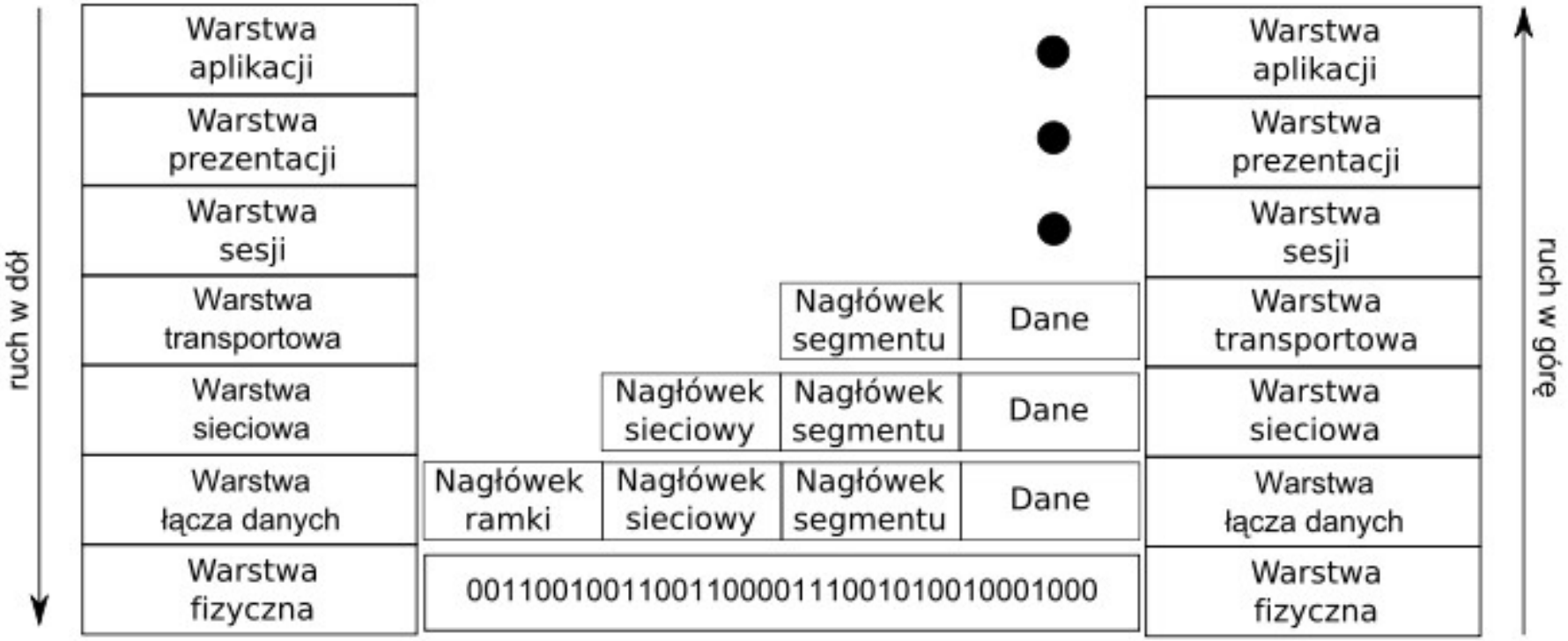


- Length/Type – znaczenie tego pola zależy od jego wartości numerycznej, pierwsze dwa bity są najważniejsze i określają czy wartość jest długością ramki lub jej typem. Jeżeli jest mniejsza lub równa 1500 to określa długość ramki. Jeżeli jest większa lub równa 1536 to określa typ ramki. Wartości pośrednie są niezdefiniowane.
- Data – pole jest sekwencją od 0 do 1500 bajtów o dowolnej wartości. W celu zapewnienia zgodności ze standardem IEEE wymagana jest minimalna długość ramki wynosząca 46 bajtów. Jeżeli wielkość przesyłanej danej jest mniejsza to w module MAC automatycznie dodawane są dodatkowe bity. Dodawanie bitów może być zablokowane za pomocą rejestru.

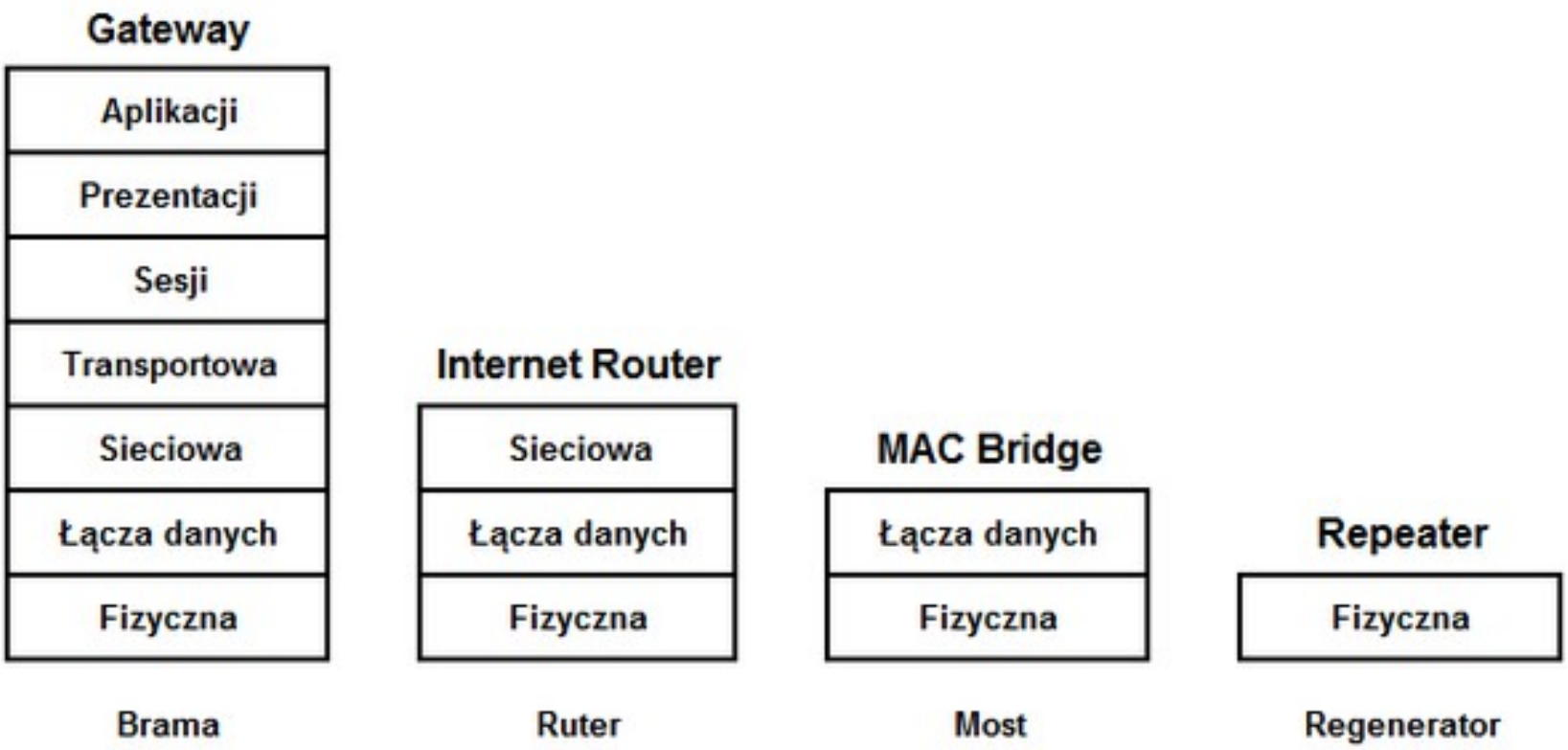
Model OSI

- OSI (ang. Open System Interconnection) lub Model OSI (model odniesienia łączenia systemów otwartych) - standard zdefiniowany przez ISO oraz ITU-T opisujący strukturę komunikacji sieciowej.
- Dla Internetu sformułowano uproszczony Model TCP/IP, który ma tylko 4 warstwy.

Kapsułkowanie danych



Zasięg pakietów w urządzeniach



Model DoD

- Warstwami dla modelu DoD (US Department of Defense) są:
  - warstwa aplikacji - obejmuje protokoły HTTP, SMTP, FTP, NFS, NIS, LPD, Telnet, SSH. Protokoły warstwy aplikacji zawierają się jako dane w protokołach warstwy transportowej.
  - warstwa transportowa – obejmuje protokoły UDP i TCP. Pierwszy dostarcza pakiety prawie bez sprawdzania poprawności transmisji, drugi natomiast gwarantuje bezstratne ich dostarczenie. Ramki warstwy transportowej zawierają się jako dane w protokole IP z warstwy sieciowej.
  - warstwa sieciowa – zawiera protokoły ICMP, IP, IGMP, RIP, OSPF i EGP. Protokół IP odpowiada za odnalezienie adresata danych w sieci. Pakiety tych protokołów są transportowane przez protokoły z warstwy łącza.
  - warstwa łącza - zawiera protokoły ARP i RARP obsługujące niskopoziomową transmisję pakietów

ARP, RARP

- **ARP** (ang. Address Resolution Protocol) - w sieciach komputerowych jest to metoda znajdowania adresu sprzętowego hosta, gdy dany jest adres warstwy sieciowej. Zdefiniowany został w RFC 826.
- Jest wykorzystywany przy różnych typach sieci, zarówno w znaczeniu warstwy sieciowej, jak i niższych warstw modelu OSI.
- ARP w sieci Ethernet jest wykorzystywany przez protokół IPv4, gdzie na podstawie adresu IP odnajduje sprzętowy adres MAC.

- **RARP** (ang. Reverse Address Resolution Protocol) Protokół komunikacyjny przekształcania 48-bitowych fizycznych adresów MAC na 32-bitowe adresy IP w komputerowych sieciach typu Ethernet.

Protokoły TCP/IP

- **TCP/IP** (ang. Transmission Control Protocol / Internet Protocol) jest pakietem najbardziej rozpowszechnionych protokołów komunikacyjnych współczesnych sieci komputerowych. Następca protokołu NCP.
- Najczęściej obecnie wykorzystywany standard sieciowy, stanowiący podstawę współczesnego Internetu.
- Nazwa pochodzi od dwóch najważniejszych jego protokołów: TCP oraz IP.
- Protokoły TCP i IP łącznie zarządzają przepływem większości danych przez sieć.
- **IP** odpowiada za przesyłanie dowolnych danych z punktu do punktu i wykorzystywany jest przez protokoły TCP lub UDP.

Protokoły TCP/IP

- Zadaniem TCP jest:
  - uzgadnianie tożsamości (handshake)
  - zarządzanie pakietami (mogą docierać do adresata w innej kolejności, niż były wysłane)
  - sterowanie przepływem
  - wykrywanie i obsługę błędów
- Para TCP+IP jest stosowana do tzw. transmisji połączeniowej, gdzie zagwarantowany jest przepływ danych dowolnej długości w obydwie strony, lub zwrotne poinformowanie nadawcy o nieusuwalnym błędzie.
- Para protokołów UDP+IP jest najczęściej używanym standardem do tzw. transmisji bezpołączeniowej, czyli przesyłania w jedną stronę, bez potwierdzania odbioru, niewielkich paczek danych zwanych datagramami.

Port Ethernet

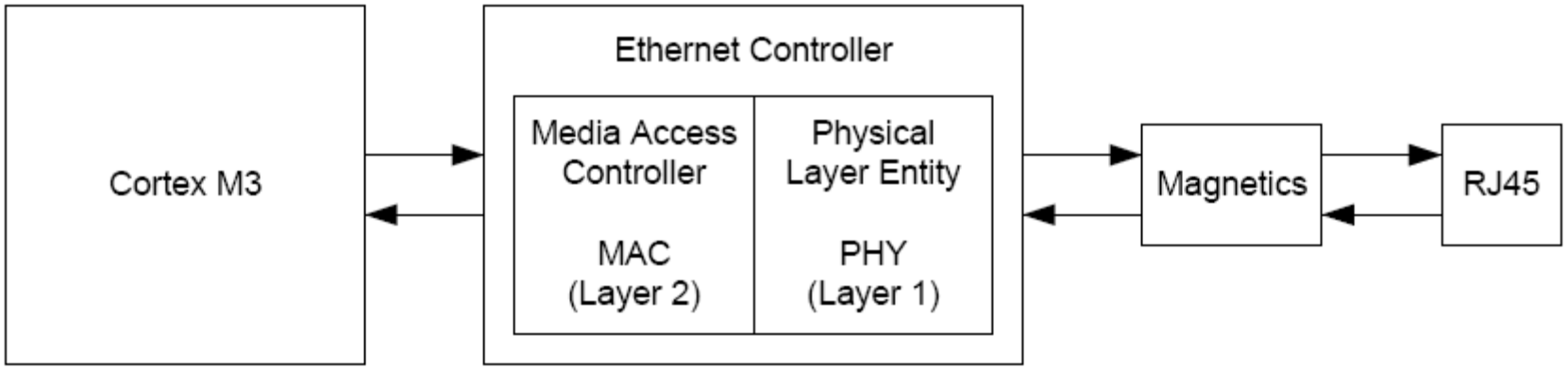
- Kontroler Ethernetu wbudowany w mikrokontrolery firmy Luminary Micro jest w pełni zintegrowanym kontrolerem składającym się ze sterownika sterującego dostępem do medium transmisyjnego (MAC) oraz interfejsu fizycznego (PHY).
- Kontroler jest zgodny z normą IEEE 802.3 i w pełni wspiera specyfikacje 10BASE-T oraz 100BASE-TX.

Port Ethernet

- Kontroler posiada następujące właściwości:
  - do pełnej realizacji obsługi Ethernetu wymaga tylko dwóch transformatorów separujących o przełożeniu 1:1,
  - umożliwia pracę w pełnym lub pół-dupleksie z prędkościami 10Mbps i 100Mbps
  - w pełni wspiera auto-negocjację szybkości transmisji,
  - programowalny adres MAC,
  - przerwania programowalne przez użytkownika,
  - kontrola CRC,
  - automatyczna detekcja i korekcja skrosowanych przewodów (MDI/MDI-X),
  - automatyczna korekta polaryzacji,
  - programowalna za pomocą rejestrów amplituda.

Port Ethernet

- Kontroler Ethernetu jest funkcjonalnie podzielony na dwie warstwy składające się z modułów:
- Media Acces Controller (MAC),
- Network Physical (PHY).
- Warstwy te odpowiadają warstwom 1 i 2 modelu OSI. Warstwa MAC odpowiada za przetwarzanie przychodzących i wysyłanych ramek, zawiera także interfejs do modułu PHY poprzez Media Independent Interface (MII).



Port Ethernet - Operacje modułu MII

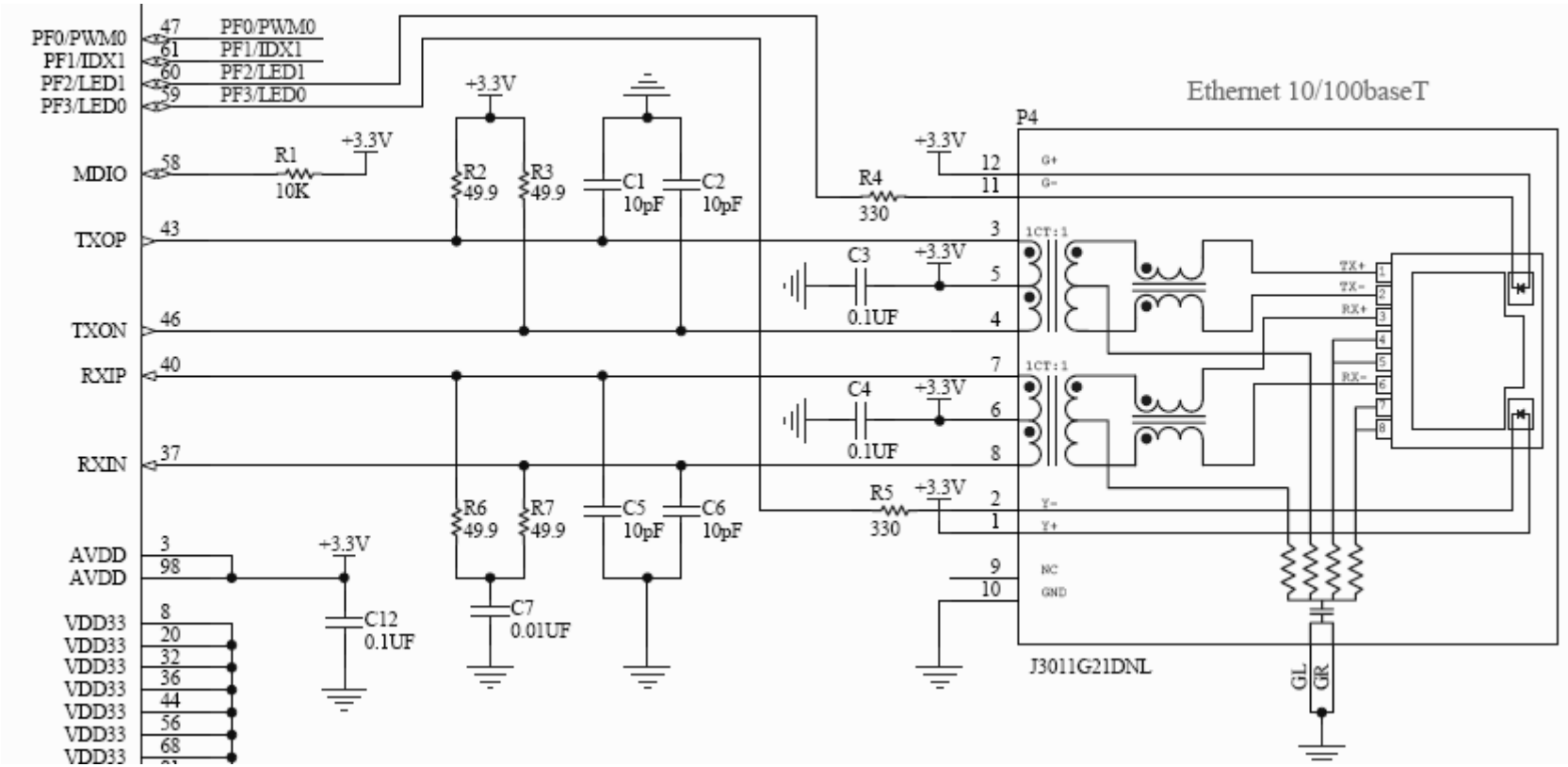
- W celu zapewnienia poprawnego funkcjonowania modułu MII konieczne jest podłączenie sygnału MDIO poprzez rezystor 10kΩ do napięcia zasilającego +3.3V.

- Dodatkowo niezbędne jest ustawienie wewnętrznego zegara na częstotliwość nie większą niż 2.5 MHz poprzez ustawienie Rejestr MACMDV,który zawiera dzielnik używany do skalowania zegara systemowego.

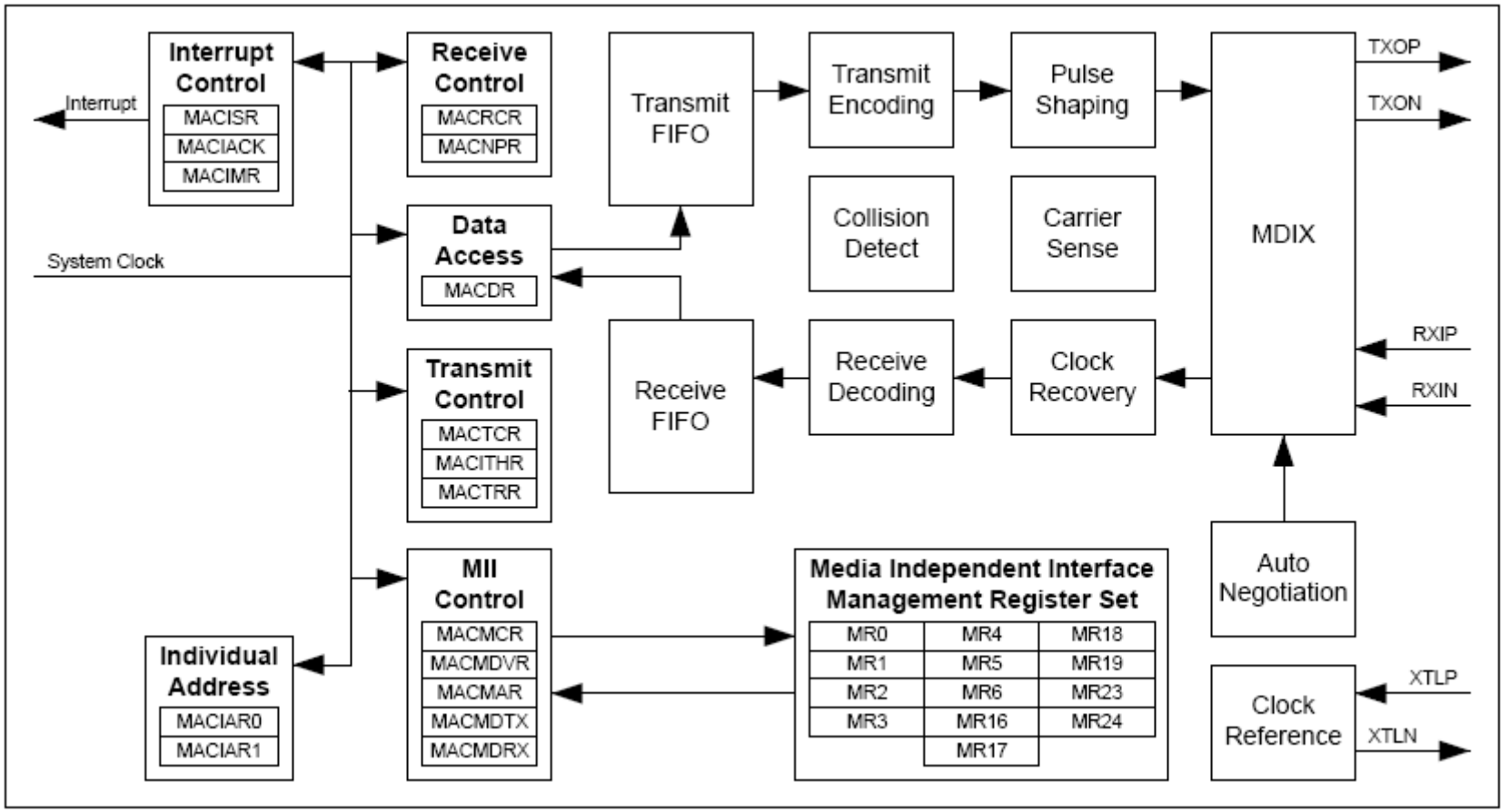
Port Ethernet - Moduł PHY

- Moduł PHY pozwala na wykorzystanie nieekranowanej skrętki Kategorii-5 dla 100BASE-TX i skrętki Kategorii-3 dla 10BASE-T
- Moduł PHY ma wbudowany oscylator lub może być sterowany z zewnętrznego oscylatora. W przypadku korzystania z zewnętrznego oscylatora pomiędzy wyprowadzenia XTALPPHY i XTALNPHY należy podłączyć kwarc o częstotliwości 25 MHz. Jeżeli wykorzystywany jest zewnętrzny oscylator sygnał 25 MHz należy podłączyć do wejścia XTALPPHY, pozostałe wejście należy podłączyć do masy.
- Moduł PHY umożliwia wykorzystanie dwóch diod LED do sygnalizacji różnych stanów pracy kontrolera Ethernetu. Sygnały te są mapowane na wyprowadzenia LED0 i LED1. Domyślnie wyprowadzenia te są wykorzystane jako porty PF3 i PF2 modułu GPIO. Jeżeli mają być wykorzystane w kontrolerze Ethernetu to należy przeprogramować funkcje wyprowadzeń w rejestrze MR23.

Podłączenia zewnętrzne



Schemat blokowy kontrolera



Ethernet API

- Zestaw funkcji Ethernet API zawiera niezbędne funkcje pozwalające zrealizować komunikację sterowaną przerwaniami. Funkcje te umożliwiają:
  - konfigurację i sterowania modułem MAC,
  - dostęp do rejestrów modułu PHY,
  - wysyłanie i odbiór pakietów,
  - konfigurację i sterowanie przerwaniami.
- Definicje funkcji Ethernet API zawarte są w pliku src/ethernet.h, a implementacja w pliku src/ethernet.h.

Ethernet API

- unsigned long EthernetConfigGet (unsigned long ulBase)
- void EthernetConfigSet (unsigned long ulBase, unsigned long ulConfig)
- void EthernetDisable (unsigned long ulBase)
- void EthernetEnable (unsigned long ulBase)
- void EthernetInitExpClk (unsigned long ulBase, unsigned long ulEthClk)
- void EthernetIntClear (unsigned long ulBase, unsigned long ulIntFlags)
- void EthernetIntDisable (unsigned long ulBase, unsigned long ulIntFlags)
- void EthernetIntEnable (unsigned long ulBase, unsigned long ulIntFlags)
- void EthernetIntRegister (unsigned long ulBase, void (#pfnHandler)(void))
- unsigned long EthernetIntStatus (unsigned long ulBase, tBoolean bMasked)
- void EthernetIntUnregister (unsigned long ulBase)
- void EthernetMACAddrGet (unsigned long ulBase, unsigned char #pucMACAddr)
- void EthernetMACAddrSet (unsigned long ulBase, unsigned char #pucMACAddr)
- tBoolean EthernetPacketAvail (unsigned long ulBase)

Ethernet API

- long EthernetPacketGet (unsigned long ulBase, unsigned char #pucBuf, long lBufLen)
- long EthernetPacketGetNonBlocking (unsigned long ulBase, unsigned char #pucBuf, long lBufLen)
- long EthernetPacketPut (unsigned long ulBase, unsigned char #pucBuf, long lBufLen)
- long EthernetPacketPutNonBlocking (unsigned long ulBase, unsigned char #pucBuf, long lBufLen)
- unsigned long EthernetPHYRead (unsigned long ulBase, unsigned char ucRegAddr)
- void EthernetPHYWrite (unsigned long ulBase, unsigned char ucRegAddr, unsigned long ul-
- Data)
- tBoolean EthernetSpaceAvail (unsigned long ulBase)

Ethernet API

- W przypadku większości aplikacji funkcja EthernetInitExpClk() musi być wywoływana jako pierwsza, w celu przygotowania kontrolera Ethernet do działania. Ta funkcja konfiguruje kontroler bazując na parametrach systemu takich jak szybkość zegara.
- Po zainicjowaniu dostęp do rejestrów w module PHY możliwy jest poprzez wykorzystanie funkcji EthernetPHYRead() and EthernetPHYWrite().
- Domyślnie PHY ustawiony jest w trybie automatycznej negocjacji prędkości i trybu komunikacji (duplex modes). Dla większości aplikacji ustawienia te są wystarczające. Jeżeli wymagana jest specjalna konfiguracja to należy wykorzystać funkcje zapisu i odczytu dla modułu PHY.

Ethernet API

- Moduł Ethernet MAC należy skonfigurować przez wywołanie funkcji EthernetConfigSet(). Parametry tej funkcji pozwalają ustawić różne tryby pracy modułu (Promiscuous Mode, Multicast Reception, Transmit Data Length Padding, itd.). Funkcja EthernetConfigGet() wykorzystywana jest do odczytu aktualnej konfiguracji modułu MAC.
- Adres MAC wykorzystywany do filtracji przychodzących pakietów ustawiany jest za pomocą funkcji EthernetMacAddrSet(), odczyt ustawionego adresu realizuje funkcja EthernetMACAddrGet().
- Kiedy konfiguracja jest zakończona należy odblokować kontroler za pomocą funkcji EthernetEnable().
- W celu wyłączenia kontrolera należy wywołać funkcję EthernetDisable().

Ethernet API

- Kiedy kontroler jest odblokowany ramki Ethernet mogą być transmitowane i odbierane przy wykorzystaniu funkcji EthernetPacketPut() i EthernetPacketGet(). Należy uważać przy wykorzystaniu tych funkcji, ponieważ są to funkcje blokujące i nie wychodzą z wywołania dopóki dane nie są dostępne przy odbiorze (RX) lub bufor ma przestrzeń do zapisu przy nadawaniu (TX). Przed użyciem funkcji blokujących, funkcje EthernetSpaceAvail() i EthernetPacketAvail() mogą być wywoływane do określenia, czy jest wolna przestrzeń bufora oraz czy są pakiety w buforze odbiorczym.
- Innym sposobem obsługi jest wykorzystanie funkcji EthernetPacketGetNonBlocking() i EthernetPacketPutNonBlocking(), które powracają z wywołania natychmiast po sprawdzeniu, czy są dostępne nowe dane lub czy w buforze jest wolna przestrzeń.

Ethernet API

- W przypadku tworzenia programu z wykorzystaniem TCP/IP należy wykorzystać możliwości generowania przerwań przez kontroler Ethernet.
- Zarejestrowanie przerwania możliwe jest w sposób statyczny poprzez zadeklarowanie funkcji w tabeli obsługi przerwań lub dynamicznie poprzez wykorzystania funkcji EthernetIntRegister(), EthernetIntUnregister().
- Do odblokowania odpowiednich przerwań służą funkcje EthernetIntEnable(), EthernetIntDisable().
- Do odpytywania i kasowania przerwań służą funkcje EthernetIntStatus() i EthernetIntClear().

Przykłady obsługi

```
unsigned char pucMACAddress[6];
unsigned char pucMyRxPacket[];
unsigned char pucMyTxPacket[];
unsigned long ulMyTxPacketLength;

//Inicjacja kontrolera  Ethernet

EthernetInitExpClk(ETH_BASE, SysCtlClockGet());

//Konfiguracja kontrolera
// Enable TX Duplex Mode
// Enable TX Padding

EthernetConfigSet(ETH_BASE, (ETH_CFG_TX_DPLXEN | ETH_CFG_TX_PADEN));

// Ustawienie adresu  MAC (01-23-45-67-89-AB)

pucMACAddress[0] = 0x01;
pucMACAddress[1] = 0x23;
pucMACAddress[2] = 0x45;
pucMACAddress[3] = 0x67;
pucMACAddress[4] = 0x89;
pucMACAddress[5] = 0xAB;
EthernetMACAddrSet(ETH_BASE, pucMACAddress);
```

Przykłady obsługi

```
// Odblokowanie kontrolera Ethernet

EthernetEnable(ETH_BASE);

// Wysłanie pakietu.

EthernetPacketPut(ETH_BASE, pucMyTxPacket, ulMyTxPacketLength);

// Czekanie i odczyt pakietu

EthernetPacketGet(ETH_BASE, pucMyRxPacket, sizeof(pucMyRxPacket));
```