

Pomocná aritmetická jednotka

Semestrální projekt

BPC-NDI 2022

Autor: Vojtěch Dvořák

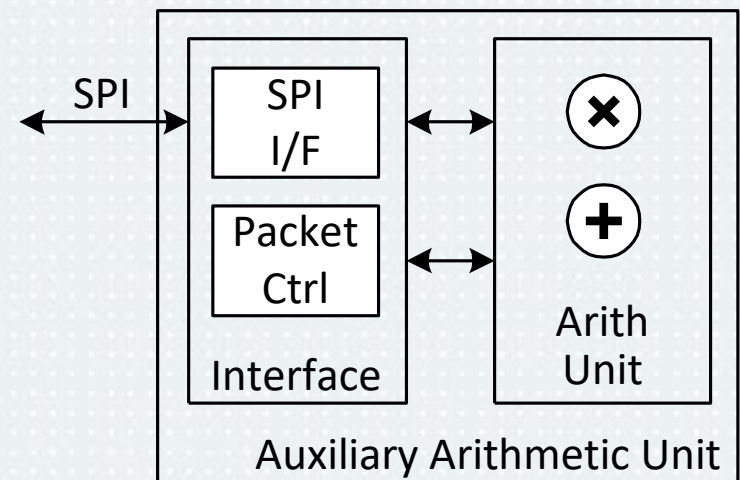
12. říjen 2022

Osnova

- Představení projektu
- Rozbor požadavků (fáze návrhu - krok 1)
- Systémový návrh/návrh architektury (fáze návrhu - krok 2)
- Návrh funkčních bloků (fáze návrhu - krok 3)

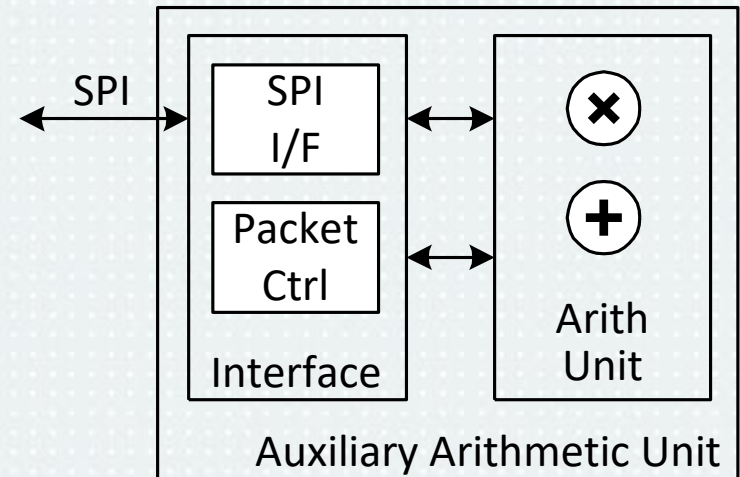
Představení projektu

- Pomocná aritmetická jednotka (AAU) provádějící elementární aritmetické operace (sčítání a násobení) s čísly, která jsou poslána přes komunikační rozhraní
- Data jsou přenášena po lince SPI (AAU je *slave*)
- Komunikační protokol - rámce, pakety, kontrola chyb na lince
- Výsledkem práce
 - RTL popis
 - Verifikační prostředí
 - Výsledky implementace
 - Dokumentace – **50 bodů**



Představení projektu

- Průběh práce na projektu (dokumentace průběžně):
 - Rozbor specifikace (krok 1) ← *společně v rámci této přednášky*
 - Systémový návrh (krok 2) ← *této přednášky*
 - Návrh dílčích bloků, tvorba RTL popisu a simulace na blokové úrovni (krok 3)
 - Verifikace pomocí simulace (krok 4)
 - Implementace (krok 5)



Rozbor specifikace

- Požadavky na AAU
 - Obecné – spolehlivost, cílový obvod, dokumentace
 - Funkční – funkce AAU, především výpočty
 - Požadavky na rozhraní – definice komunikační linky a chybových stavů
 - Viz dokument AAU_RS_0001

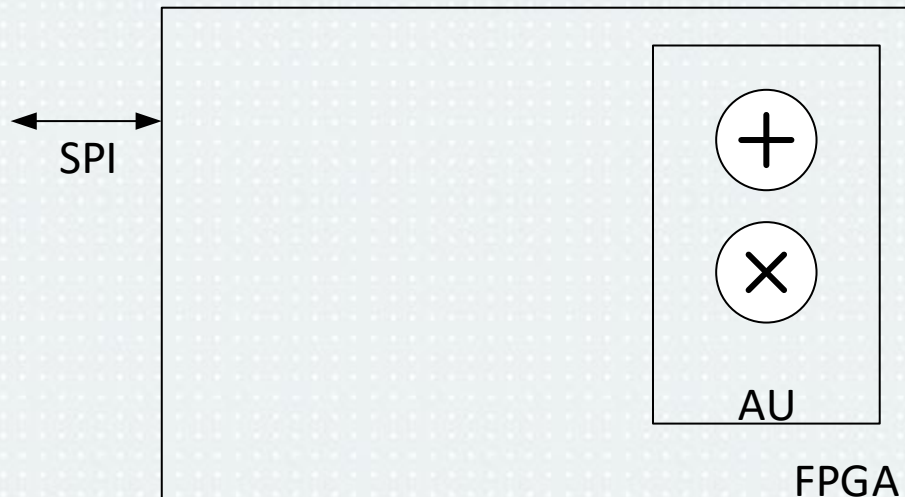
Definice architektury

- Po seznámení se specifikací - návrh obecné struktury
 - Postup top → down
- 1. Rozhraní FPGA (clk, rst implicitně součástí návrhu)



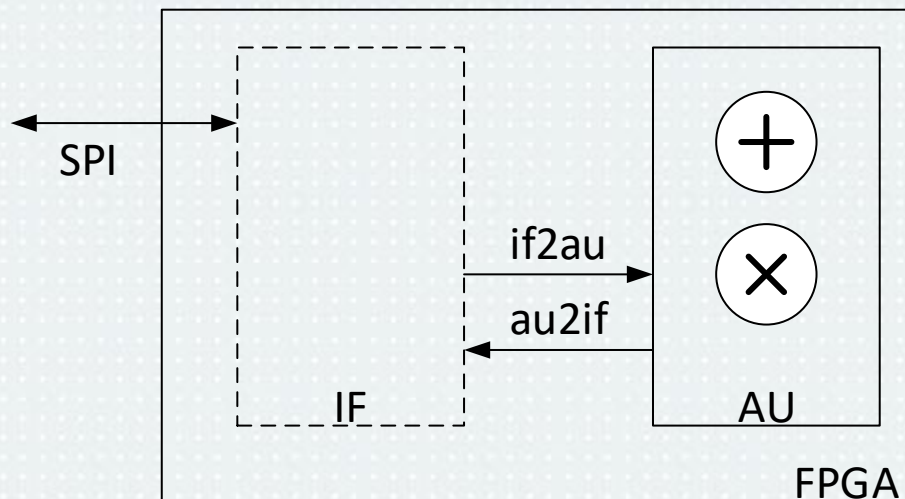
Definice architektury

- Po seznámení se specifikací - návrh obecné struktury
 - Postup top → down
2. Hlavní funkce FPGA – aritmetické operace



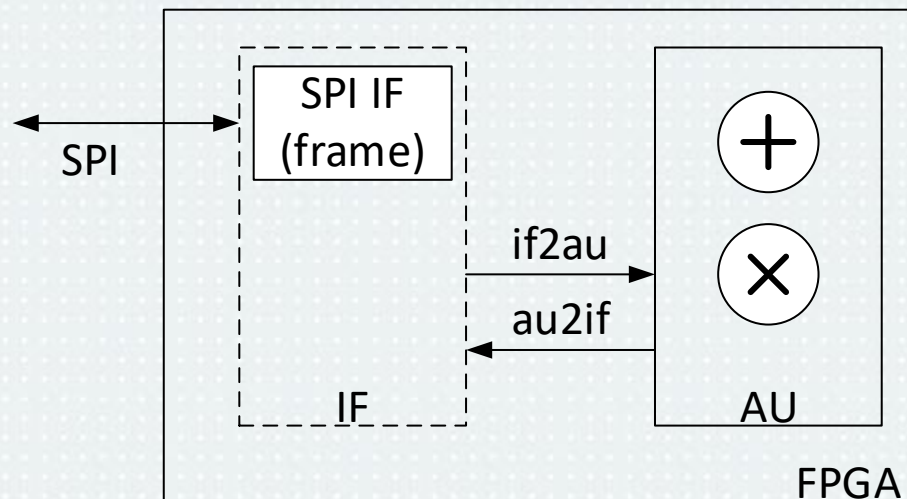
Definice architektury

- Po seznámení se specifikací - návrh obecné struktury
 - Postup top → down
- 3. Komunikace – potřeba komunikačního rozhraní s nadřazenou jednotkou (a dále do AU)



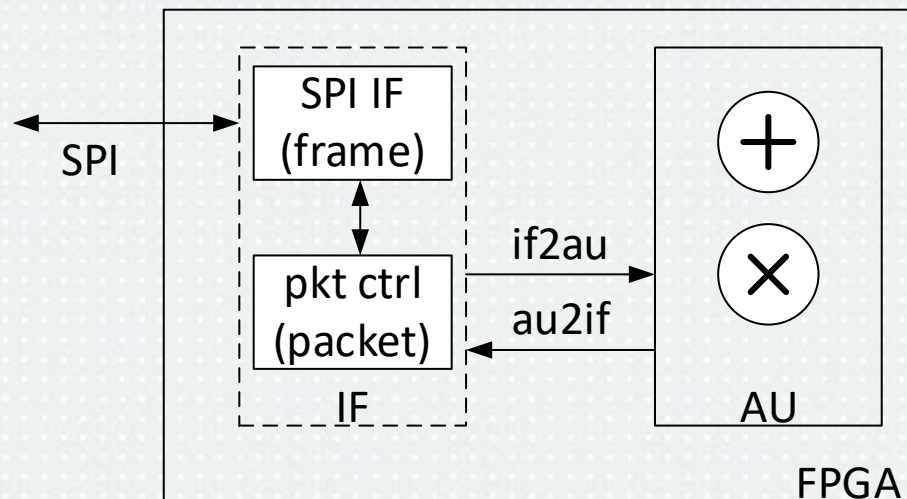
Definice architektury

- Po seznámení se specifikací - návrh obecné struktury
 - Postup top → down
- 4. Komunikace – rozdělení podle komunikační vrstvy, přenos rámců po lince SPI



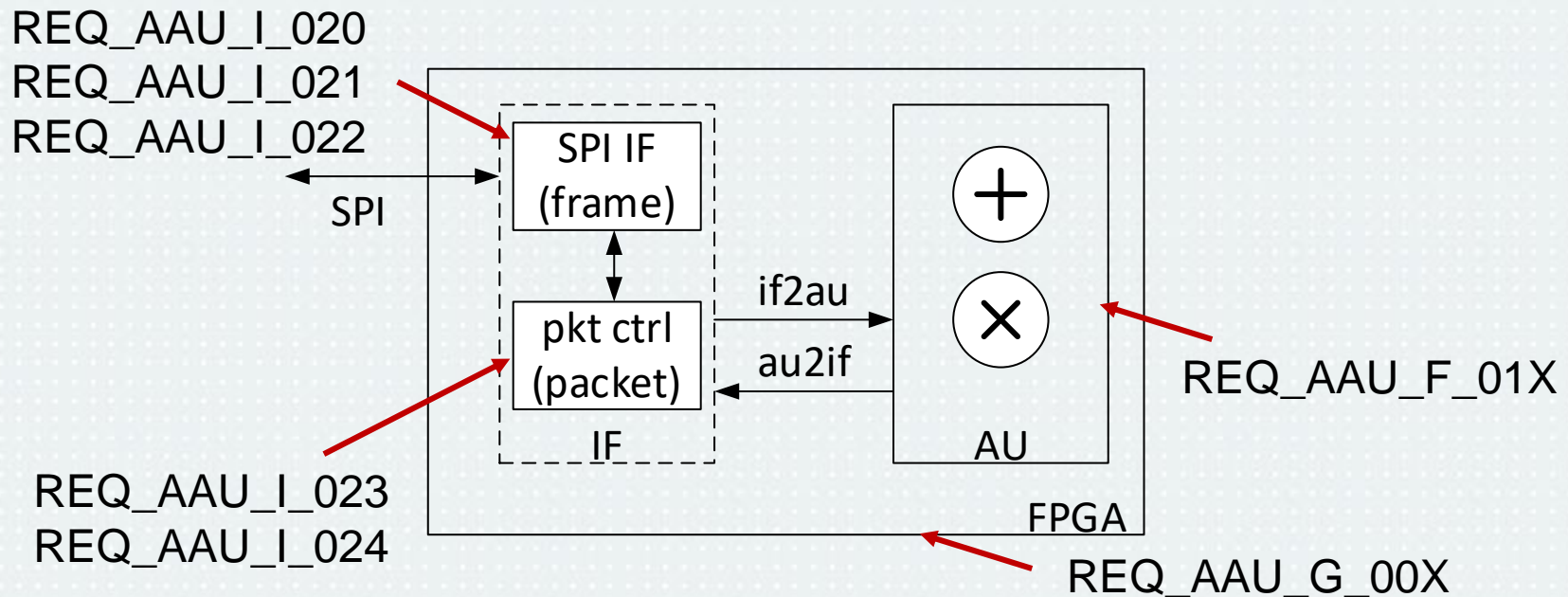
Definice architektury

- Po seznámení se specifikací - návrh obecné struktury
 - Postup top → down
- 5. Komunikace – rozdělení podle komunikační vrstvy, paket tvořen rámcem



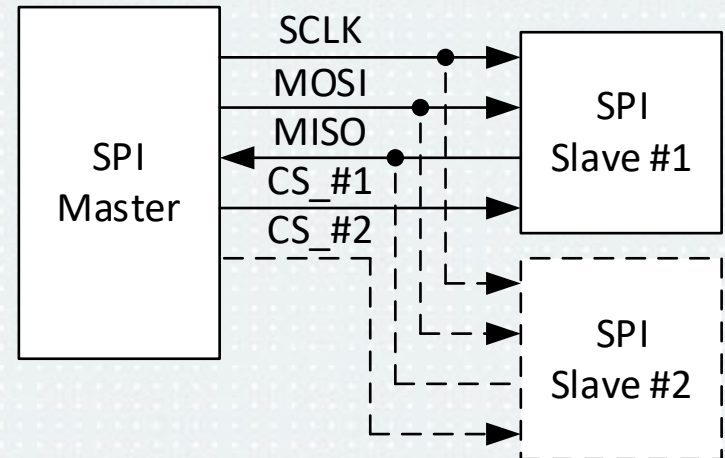
Definice architektury

- Alokace požadavků na jednotlivé funkční bloky
 - Obecné požadavky – AU i IF
 - Funkční požadavky – aritmetická jednotka (AU)
 - Požadavky na rozhraní – komunikační rozhraní (IF)



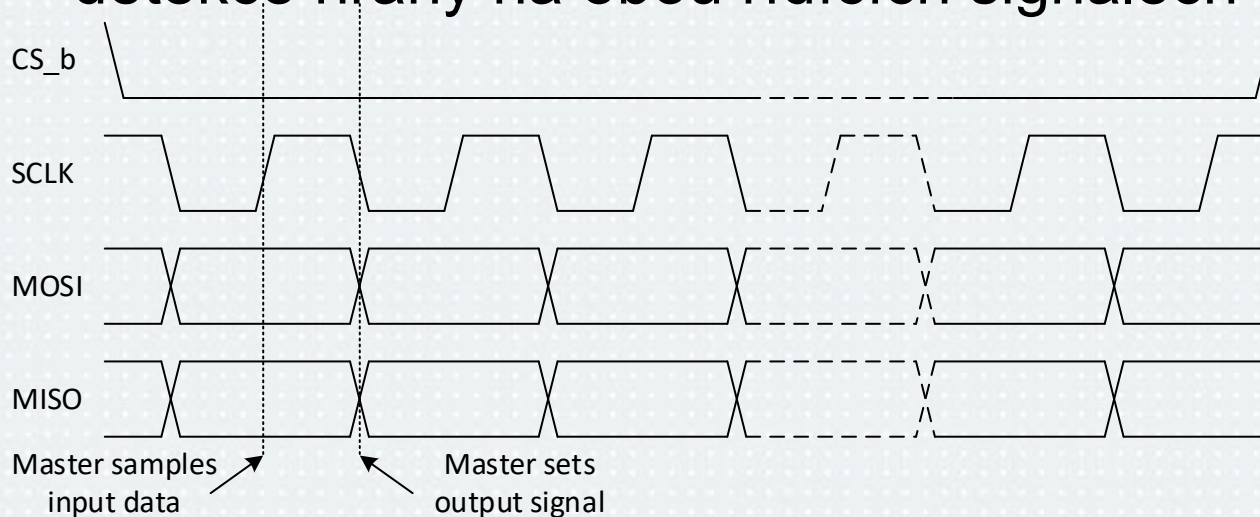
Návrh dílčích bloků

- Sběrnice SPI - sériová linka
- (obvykle) čtyři signály
 - dva řídící (SCLK, CS)
 - dva datové (MOSI, MISO)
- Dva účastníci
 - Master (vždy jen jeden)
 - Zahajuje a ukončuje komunikaci
 - Generuje oba řídící signály SCLK a CS
 - Řídí datový signál MOSI (Master-Out-Slave-In)
 - Slave (může být i více než jeden)
 - naslouchá řídícím signálům
 - Řídí datový signál MISO (Master-In-Slave-Out)



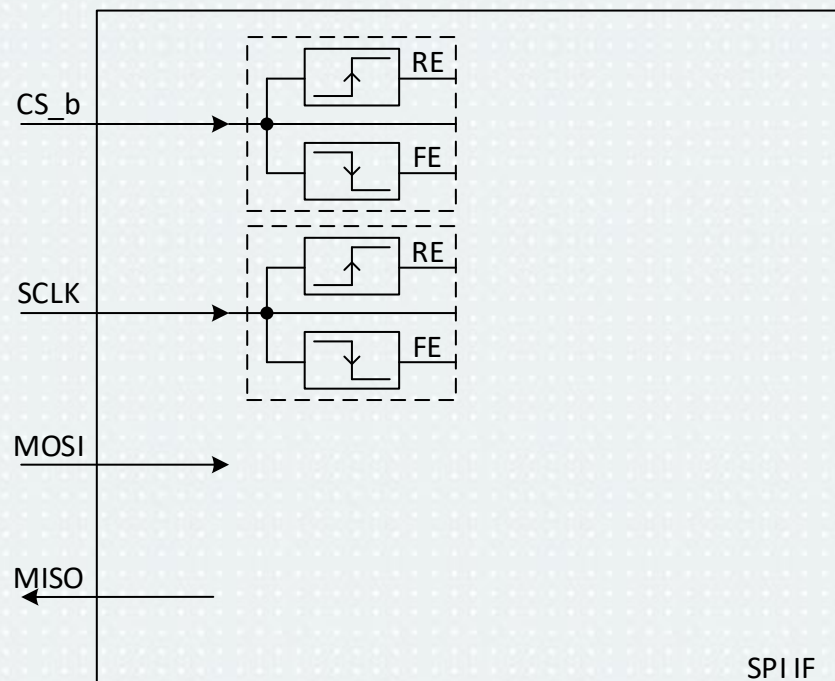
Návrh dílčích bloků

- Návrh datového rozhraní (IF) – vrstva přenosu rámců
 - Rámec definován signálem CS_b
 - Data (MOSI) nastavena se sestupnou hranou SCLK, platná na náběžnou hranu
 - Data (MISO) vzorkována s náběžnou hranou SCLK, musí být nastavena na sestupnou hranu
 - → detekce hrany na obou řídících signálech



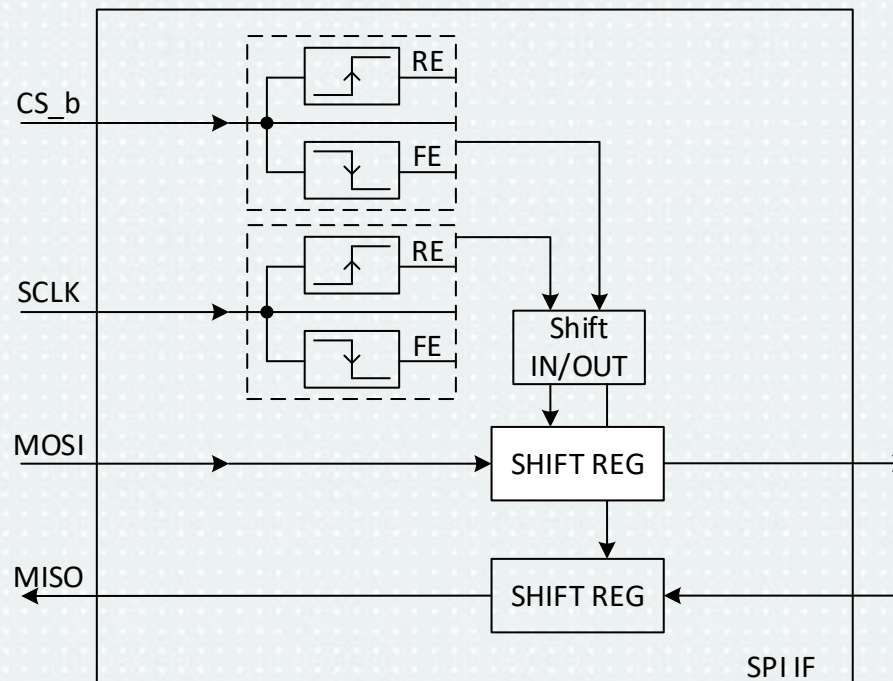
Návrh dílčích bloků

- Návrh datového rozhraní (IF) – vrstva přenosu rámců
- Vstupy a výstupy – signály SPI
- Detektor náběžné a sestupné hrany na řídicích signálech



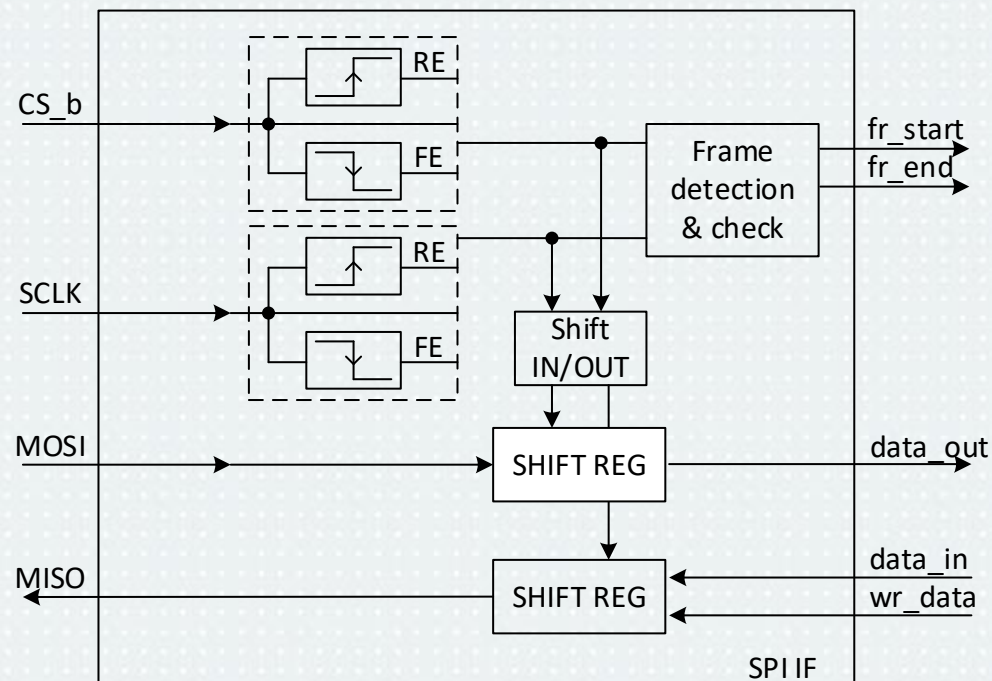
Návrh dílčích bloků

- Návrh datového rozhraní (IF) – vrstva přenosu rámců
- Datové vstupy/výstupy – převod sériové komunikace na paralelní slovo a opačně → posuvný registr
- Posun registru při detekci hrany na SCLK



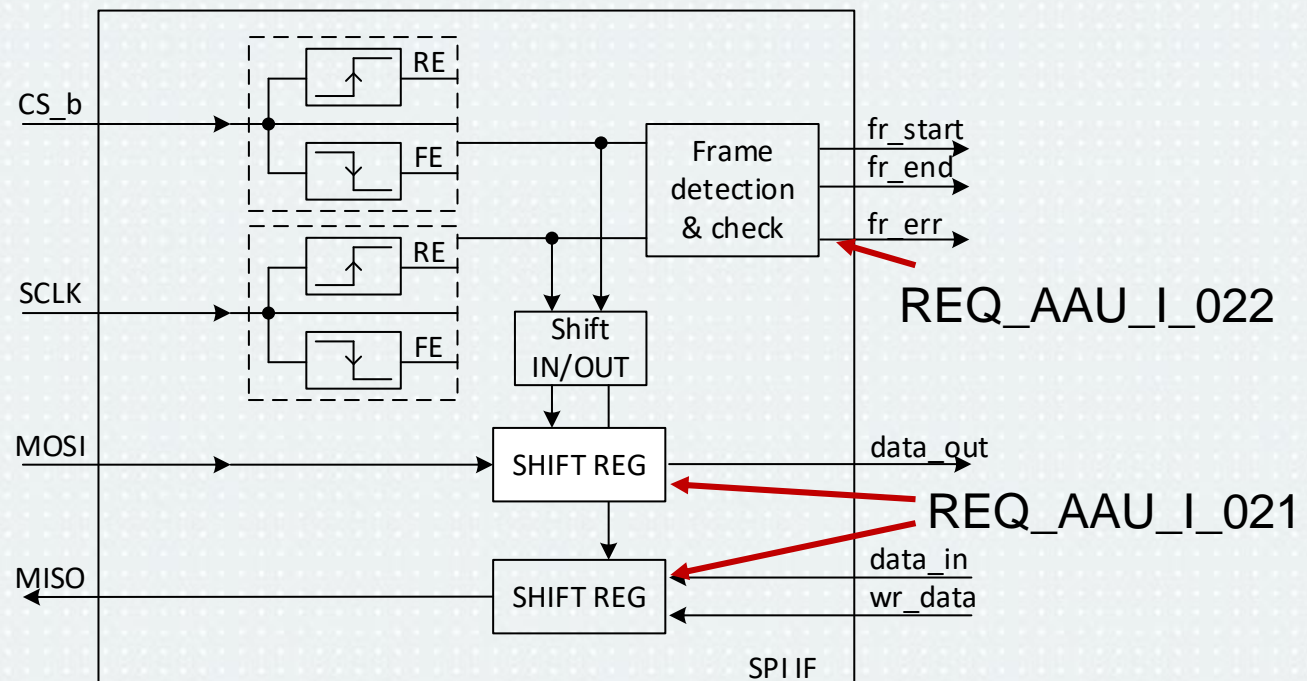
Návrh dílčích bloků

- Návrh datového rozhraní (IF) – vrstva přenosu rámců
- Detekce začátku a konce rámce



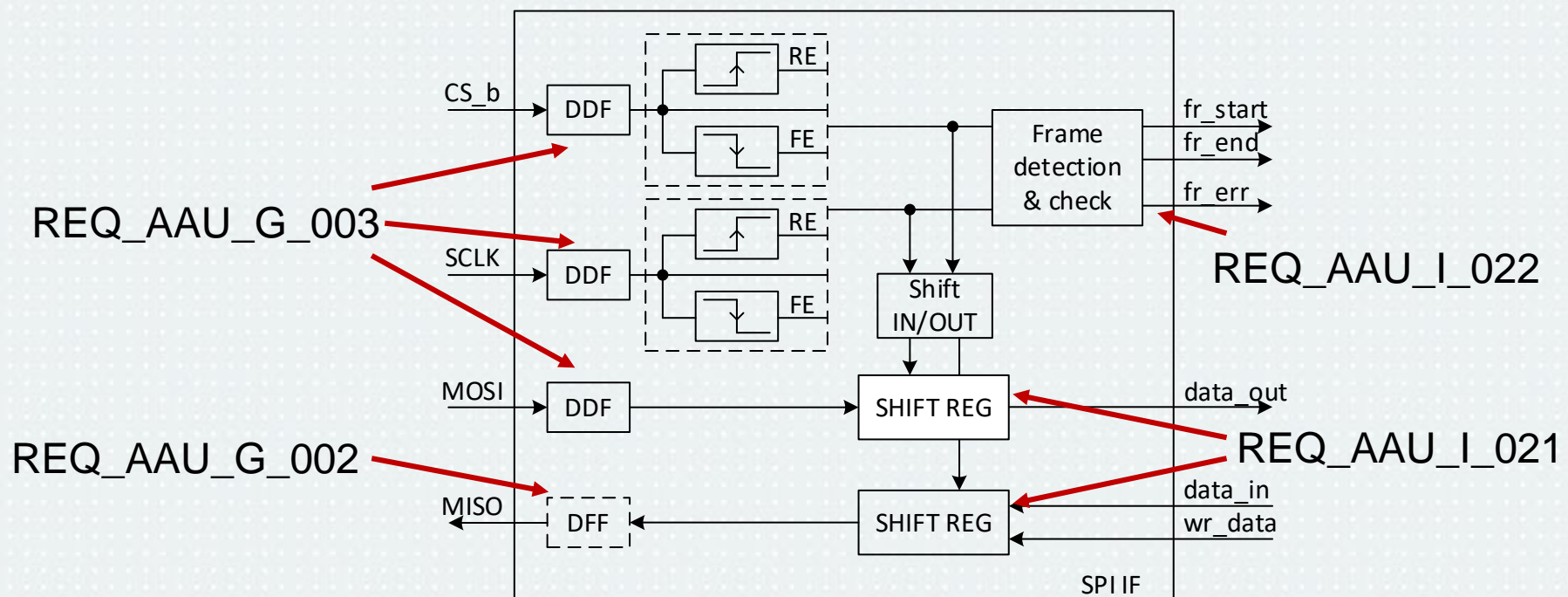
Návrh dílčích bloků

- Návrh datového rozhraní (IF) – vrstva přenosu rámců
- Požadavek na kompatibilitu se SCLK (REQ_AAU_I_020)
- Požadavek na řazení bitů (REQ_AAU_I_021)
- Požadavek na detekci chybného rámce (REQ_AAU_I_022)



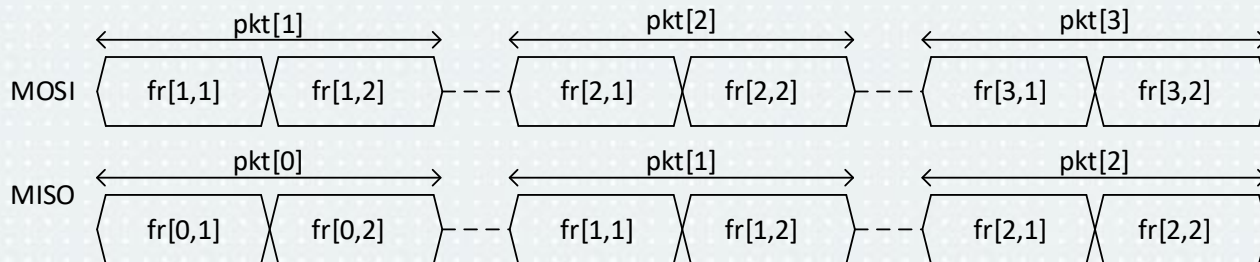
Návrh dílčích bloků

- Návrh datového rozhraní (IF) – vrstva přenosu rámců
- Požadavek na výstupní signály (REQ_AAU_G_002)
- Požadavek na vstupní signály (REQ_AAU_G_003)



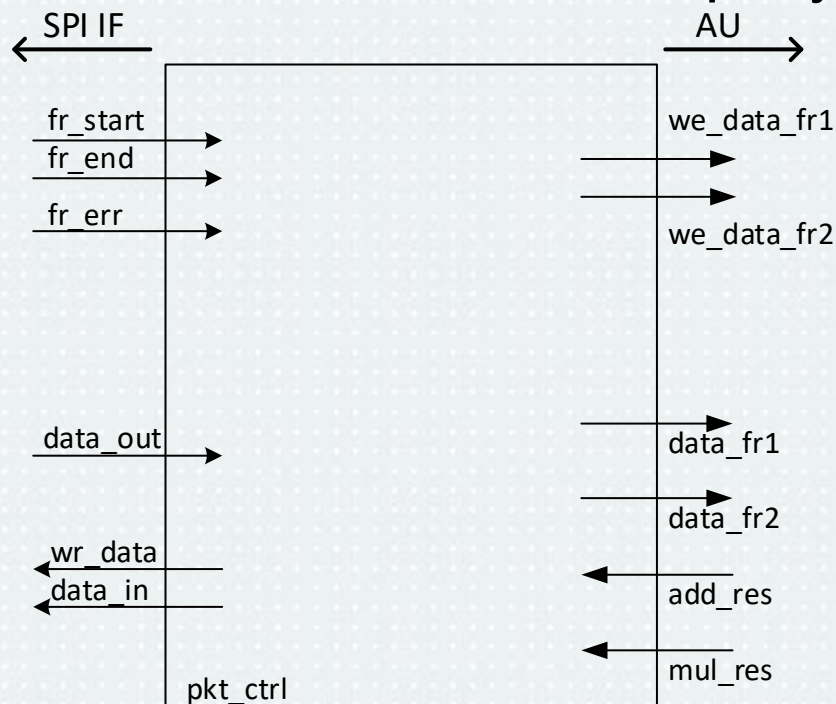
Návrh dílčích bloků

- Návrh datového rozhraní (IF) – vrstva přenosu paketů
 - Jeden paket jsou dva rámce (2 čísla)
 - Odpověď je očekávána v následujícím paketu



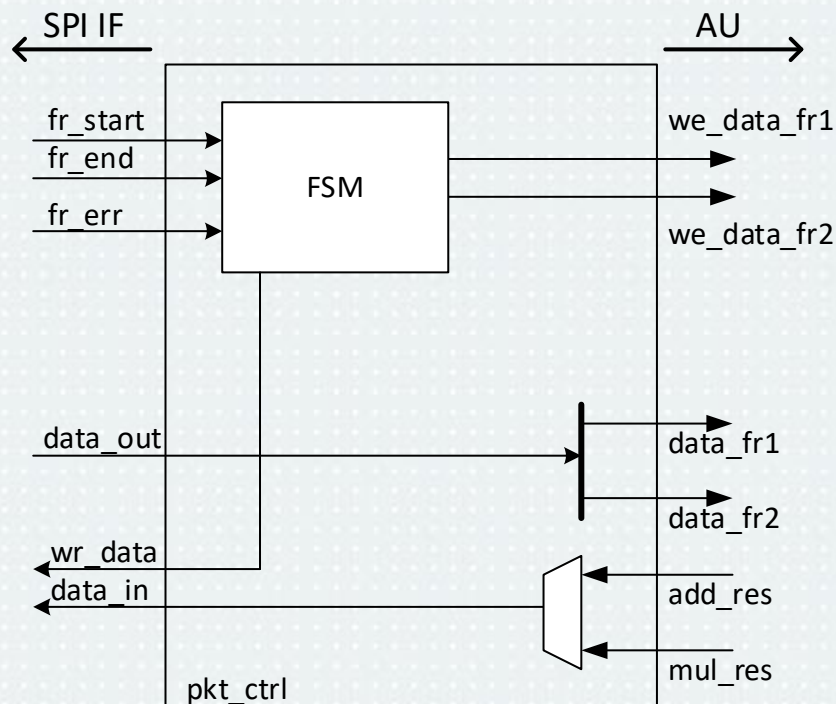
Návrh dílčích bloků

- Návrh datového rozhraní (IF) – vrstva přenosu paketů
 - Rozhraní mezi SPI IF a aritmetickou jednotkou
 - Rozhraní k SPI IF definováno výše
 - Rozhraní k AU – data a pokyn k zápisu registrů



Návrh dílčích bloků

- Návrh datového rozhraní (IF) – vrstva přenosu paketů
 - Identifikace čísla rámce v paketu a přepínání toku dat → stavový automat



Očekávané stavy

Čekání na první rámec

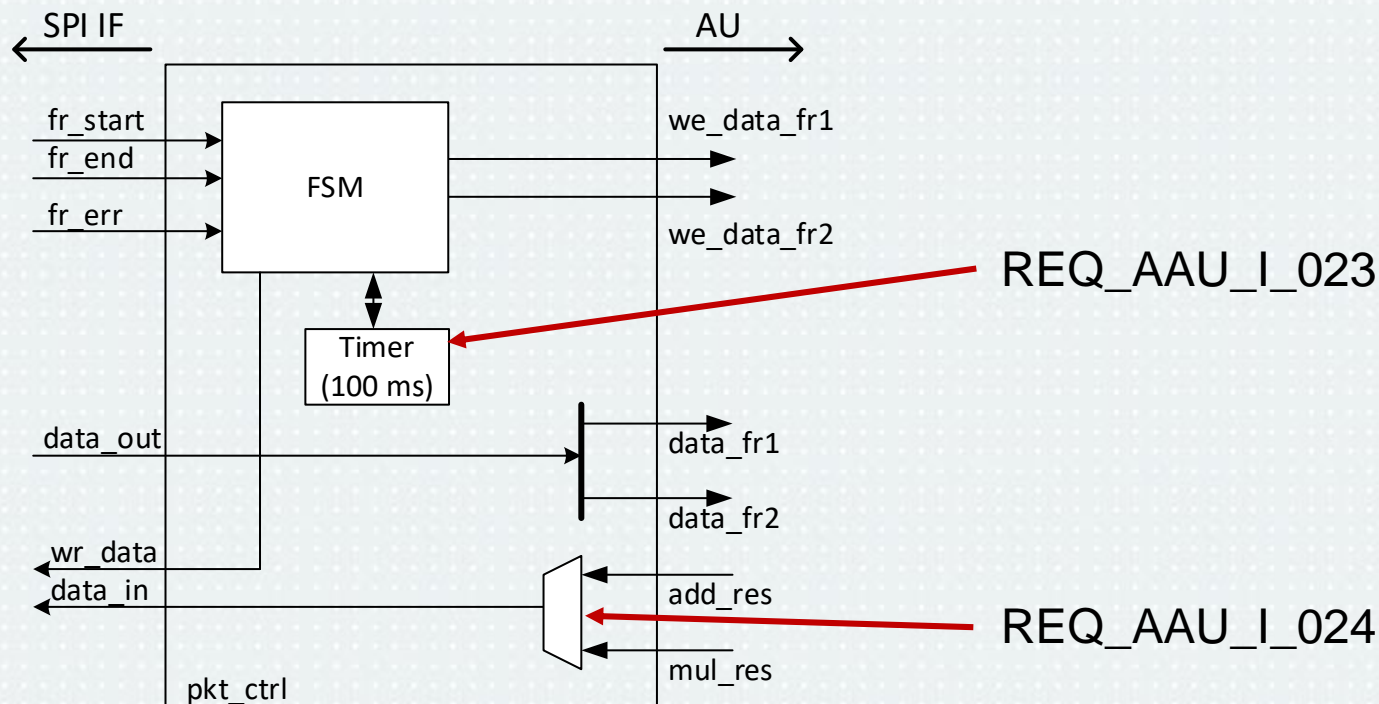
Příjem prvního rámce

Čekání na druhý rámec

Příjem druhého rámce

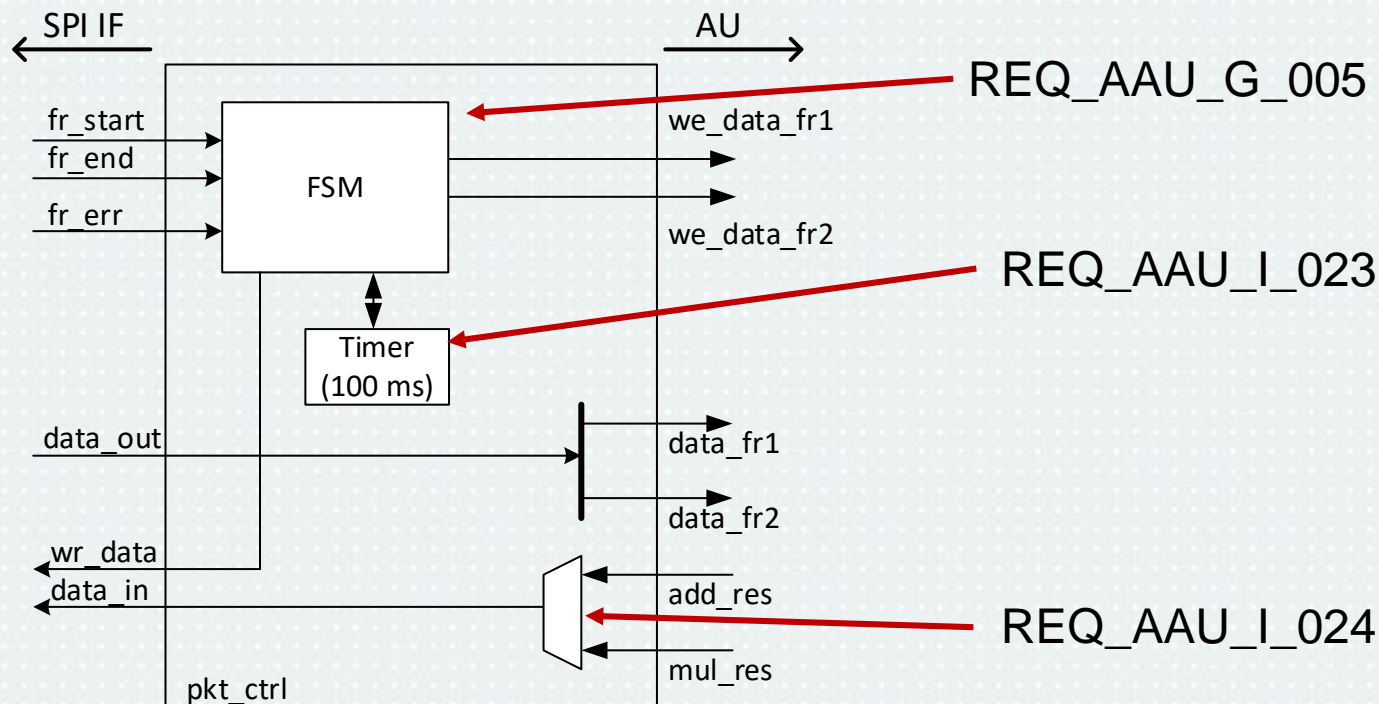
Návrh dílčích bloků

- Návrh datového rozhraní (IF) – vrstva přenosu paketů
 - Požadavek na reset linky (REQ_AAU_I_023)
 - Požadavek na formát paketů (REQ_AAU_I_024)



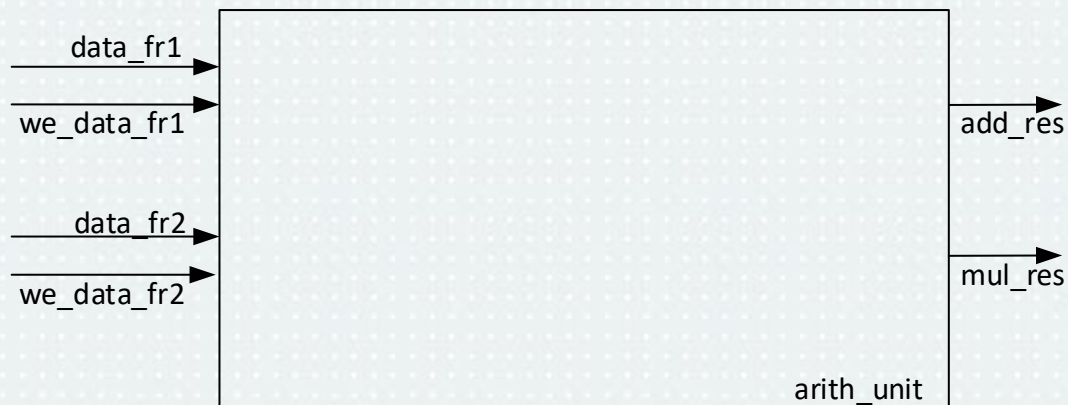
Návrh dílčích bloků

- Návrh datového rozhraní (IF) – vrstva přenosu paketů
 - Požadavek na bezpečnou implementaci FSM (REQ_AAU_G_005)



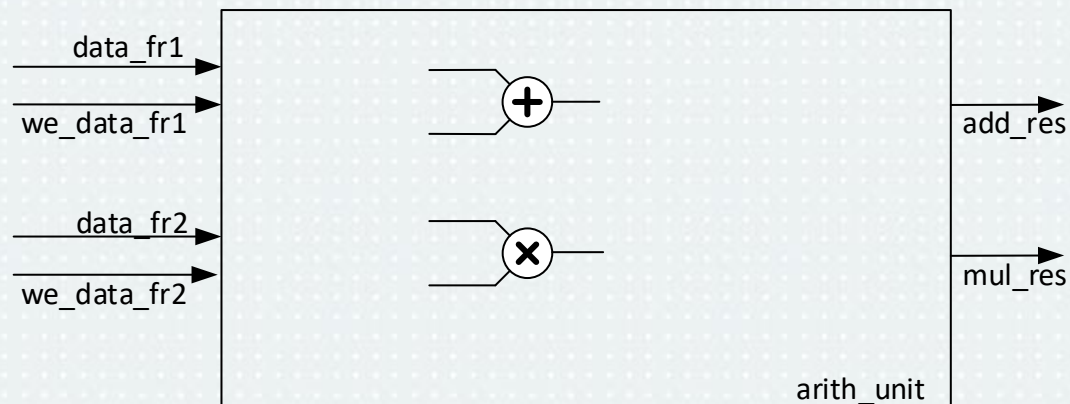
Návrh dílčích bloků

- Návrh aritmetické jednotky
 - Rozhraní k modulu pro zpracování paketů



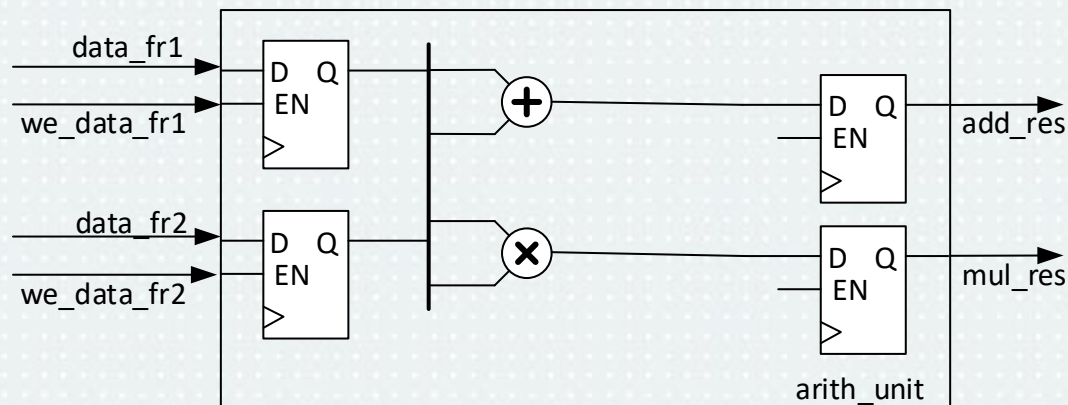
Návrh dílčích bloků

- Návrh aritmetické jednotky
 - Aritmetické operace sčítání a násobení



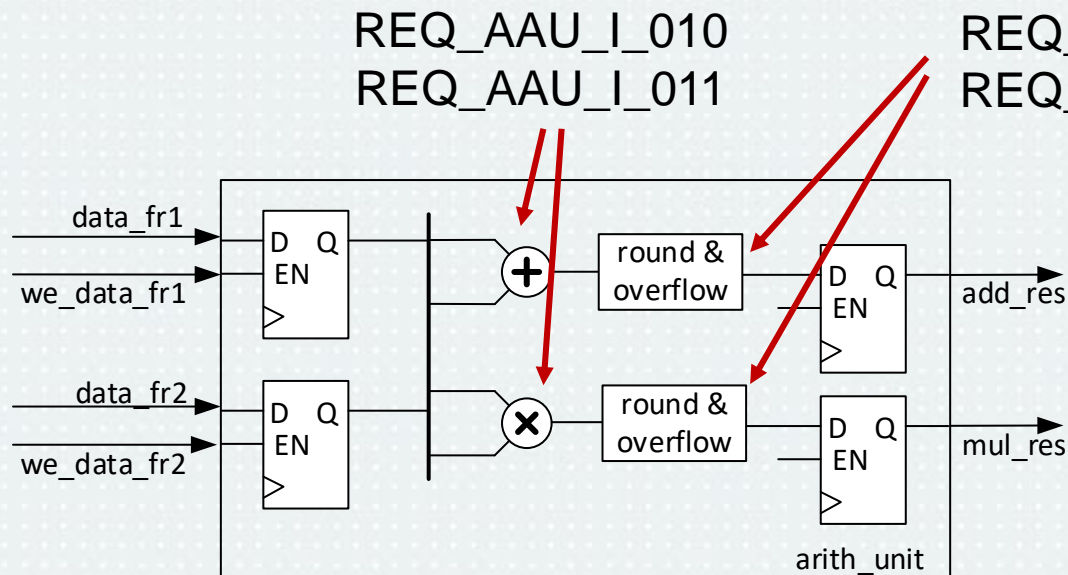
Návrh dílčích bloků

- Návrh aritmetické jednotky
 - Data přicházejí postupně – potřeba registrů pro ukládání čísel na vstupu
 - Potřeba uložit výsledek obou operací, dokud nejsou odeslány



Návrh dílčích bloků

- Návrh aritmetické jednotky
- Požadavek na aritmetické operace (REQ_AAU_F_010)
- Požadavek na formát čísel (REQ_AAU_F_011)
- Požadavek na zaokrouhlení výsledku (REQ_AAU_F_012)
- Požadavek na ošetření přetečení (REQ_AAU_F_013)

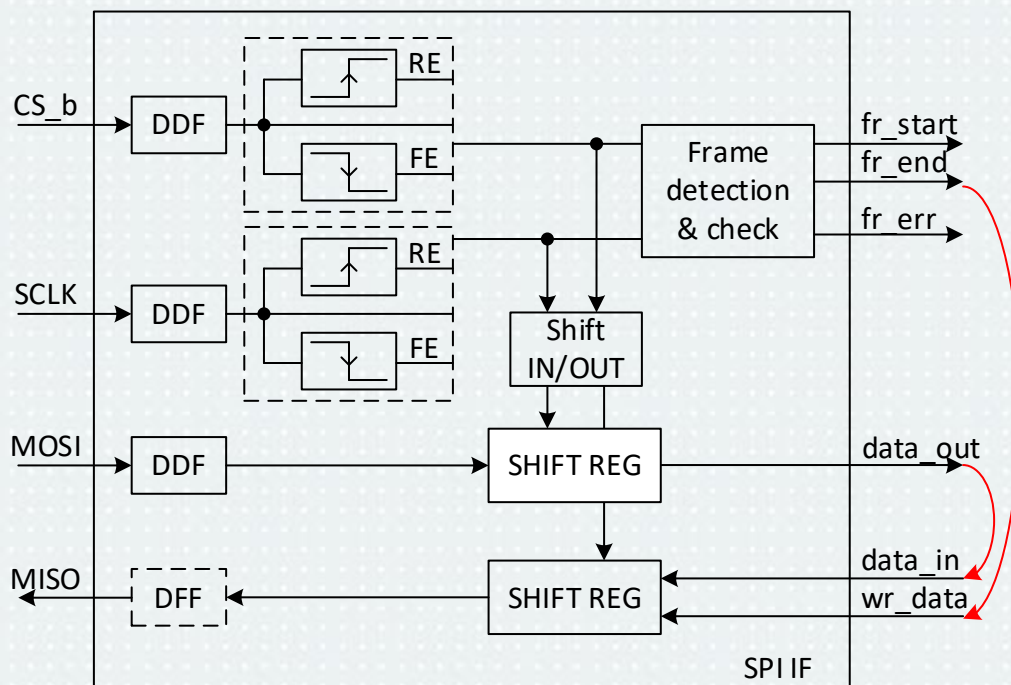


Postup práce (doporučení)

- Nejprve si prostudujte specifikaci a ptejte se
- Při návrhu postupujte blok po bloku
 - SPI IF → zpracování paketů → aritmetická jednotka
 - Doplňte chybějící kód
 - Proveďte jednoduchou simulaci pro ověření základní funkčnosti každého bloku
- Po ověření základní funkčnosti bude následovat verifikace
- Plně verifikovaný kód bude nakonec implementován
- Dokumentaci tvořte průběžně (především v průběhu verifikace)

Postup práce (doporučení)

- SPI IF
 - Detektor hran → posuvný registr → logika pro posun → detekce rámce
 - Ověření základní funkčnosti – loopback simulace



**děkuji za pozornost
(a pište komentáře)**