Zabezpečení datových přenosů pomocí CRC

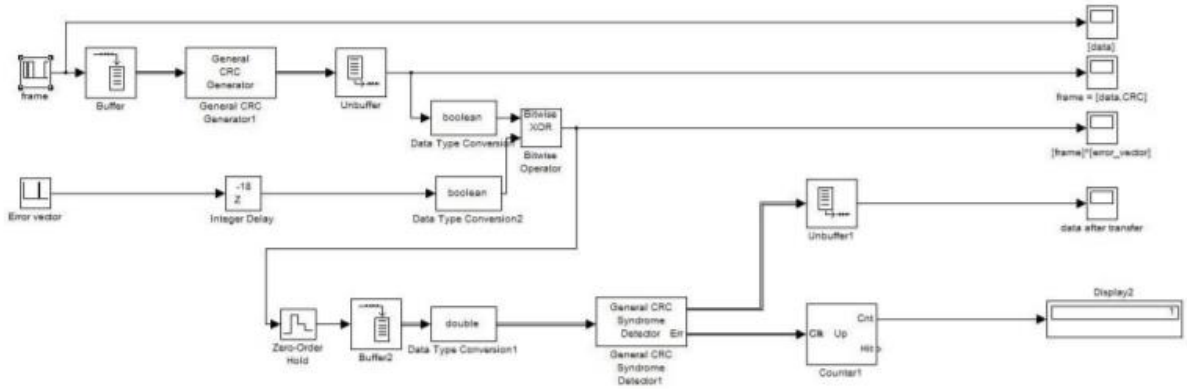
Pomocné programy:

- Matlab
- Simulink

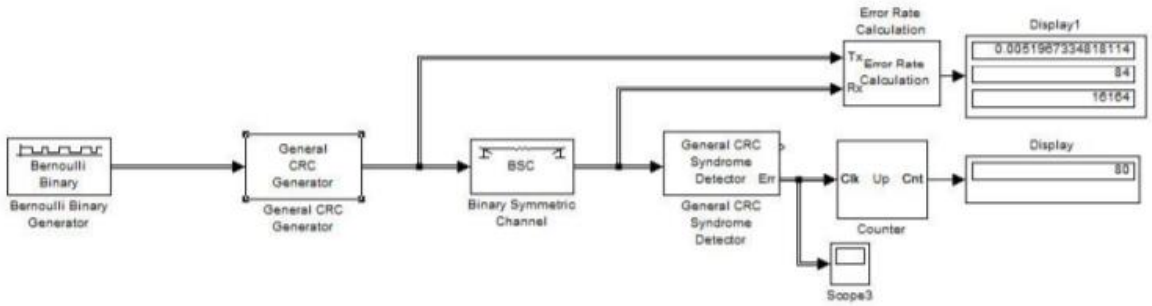
Úkoly měření:

1) Seznámit se s využitím CRC
2) Ověřit vlastnosti a detekční schopnosti CRC
3) Seznámit se s praktickou implementací CRC

Simulační model:



Obr. 1: Simulace zabezpečení pomocí CRC



Obr. 2: Simulace generování náhodných chyb v přenosovém řetězci při zabezpečení pomocí CRC

Teorie:

CRC je algoritmus výpočtu redundantních dat z dat užitečných (tedy těch, jejichž přenos chceme zabezpečit). CRC kód je odesílán spolu s daty a po jejich přijetí znovu nezávisle vypočítán. Pokud se liší vypočtený CRC kód od přijatého, je jisté, že při přenosu došlo k chybě. Pokud jsou přijatý a vypočtený CRC kód shodné, považujeme data za správně přijatá. Existuje však nízká (ale nenulová) pravděpodobnost, že přijatá data jsou chybná. Přenášenou datovou posloupnost lze interpretovat například jako polynom s binárními koeficienty. Posloupnost 11000101 lze pak reprezentovat binárním polynomem $x^7 \oplus x^6 \oplus x^2 \oplus 1$. CRC se počítá jako zbytek po dělení datové posloupnosti tzv. generujícím polynomem. V průběhu výpočtu se na koeficienty uplatňuje operace sčítání modulo 2 (tj. platí $1 \oplus 1 = 0$). Určení CRC pouze generujícím polynomem je nejednoznačné, různé algoritmy ho

implementují různě – z historických nebo technických důvodů může docházet prohození bajtů, změně pořadí bitů v bajtu, nebo přidání různých bitových posloupností před nebo za vstupní data.

Postup:

Použili jsme CRC3 a data o délce 8 bitů, konkrétně 10011101. Nastavili jsme údaje v programu Matlab a sledovali simulaci. Seznámili jsme se s výstupním formátem programu, tedy křivkou reprezentující odpovídající CRC. Vyzkoušeli jsme si různé druhy chybových polynomů. Pak jsme zkusili to samé, ale s chybovým vektorem, který by byl posunem generujícího polynomu a opravdu jsme dostali 0 chyb. Všechny případy jsme ověřili ručně pomocí papíru a tužky.

Těmito výpočty jsme ověřili následující **hypotézy:**

1. Je-li chybový vektor posunem generujícího polynomu (násobení obecnou mocninou), chyba není detekována.
2. Obecněji: je-li chybový vektor beze zbytku dělitelný generujícím polynomem, chyba není detekována.
3. Každá jednonásobná chyba je detekována, pokud má generující polynom koeficient 1 u x^0 a zároveň má alespoň jeden další člen.
4. Pokud je generující polynom beze zbytku dělitelný polynomem $x \oplus 1$, pak detekuje jakýkoliv lichý počet chyb.
5. Pokud $x^i \oplus 1$ není beze zbytku dělitelné generujícím polynomem pro všechna $i \in \langle 1, n-1 \rangle$, kde n je délka kódového slova, pak jsou detekovány všechny dvojnásobné chyby.
6. Pokud je chybový vektor typu $x^j(x^t \oplus ... \oplus 1)$, kde t je menší než stupeň generujícího polynomu (blok chyb s délkou menší nebo rovnou počtu bitů CRC), pak jsou všechny takovéto chyby detekovány.

Tvrzení: Pokud je chybový vektor blokovou chybou s délkou rovnou délce generujícího polynomu (vektor typu $x^j(x^t \oplus ... \oplus 1)$, kde t je rovno stupni generujícího polynomu), pak pravděpodobnost, že chyba nebude detekována, je $2^{-(t-1)}$.

Důkaz: Výraz $(x^t \oplus ... \oplus 1)$ je dělitelný generujícím polynomem jenom tehdy, když se rovná generujícímu polynomu. Takže bloková chyba (t + 1) chybového vektoru se rovná bloku (t + 1) v generujícím polynomu. Podle definice blok začíná a končí jedničkami, tedy závisí na (t + 1) - 2 = t - 1 bytech, které jsou uprostřed bloku. Proto je pravděpodobnost $2^{-(t-1)}$.

Tvrzení: Pokud je chybový vektor blokovou chybou s délkou vyšší, než je stupeň generujícího polynomu, pak pravděpodobnost, že chyba nebude detekována, je 2^{-t}.

Důkaz: Když přidáváme n-bytový CRC k naši zprávě, zvětšujeme celkový počet přípustných řetězců na násobek 2^t. Tudíž máme situaci, když jenom jeden řetězec z 2^n je správný. Takže za předpokladu, že zkreslení dat ovlivňuje náš řetězec náhodně, pravděpodobnost toho, že chyba nebude detekována, je 2^{-t}.

Spolehlivost:

Jako generující polynom jsme ve všech pokusech vybrali CRC3. Horní řádek tabulky představuje pravděpodobnost chyby:

	1%	0.1%	0.01%
Data:	1924103	11124103	11125219
Errors	119025	11163	1000
Real errors	65535	10572	1064

Z toho jsme usoudili, že polynom nízkého stupně je spolehlivý pouze v sítích, kde je pravděpodobnost chybného přenosu bitu velmi nízká.

Závěr:

Pomocí simulátoru jsme si ověřili princip CRC. Zjistili jsme, že řetězec s CRC daty je o jeden bit kratší, jelikož zbytek po dělení má vždy nižší stupeň než dělitel. Tedy délka odesílaného řetězce bude délka zprávy + stupeň polynomu. U simulátoru nefungovala v některých případech detekce chyb tak, jak bychom očekávali, takže jsme museli počítat na papíře. Následně jsme si ověřili pravdivost hypotéz, kdy je chybný datový přenos detekován a za jakých podmínek nemůže uzel špatný přenos objevit.