

# 1 Organizační úvod

Budeme se zabývat hlavně hardwarem + Operační systémy v letním semestru tím zbytkem. Volně navazuje i architektura počítačů v LS 2. ročníku. Chce se po nás umět souvislosti a termíny (občas budou české, ale vždy i s anglickou variantou). Budeme si lhát sofistikovaně.

## 2 Harvardská architektura

Podle univerzity

- procesor (CPU) – základní výpočetní jednotka
- kódová paměť (code memory) – paměť
- propojení – CPU přes něj čte data z paměti (měnit program (tedy to, co je v této paměti) neumí)
- datová paměť (data memory) – proměnné, lze do ni zapisovat (W) i z ní číst (R)
- input / output (I/O, tzv. periférie) – komunikace s okolím

Anglický (chudý) matematik Charles Babbage navrhl *Analytical engine* (1837, pohonem byl parní stroj, měl počítat daně), který měl tuto architekturu.

Ada Lovelace jeho manželka napsala manuál k Analytical engine (jako matematik to moc nepopsal). Zároveň vymyslela, že kromě čísel by šlo předávat i zakódovaná písmena, noty, atd.

## 3 Reprezentace (celých) čísel

### 3.1 přenos

- Můžeme reprezentovat analogově 1 jedním voltem, 2 dvěma,  $10^6$  milionů.
- Můžeme použít i jinak naškálovanou hodnotu, 1 tisícinou voltu, 2 dvěma tisícinami  $10^6$  1000 voltů. Vodiče ale nejsou ideální vodiče, mění odpor / proud podle délky, tepla a elektromagnetického pole v okolí.
- Můžeme používat digitální (číslíkový) přenos, třeba (někdy je to opačně): napětí nad nějakou hodnotou jako 1 a napětí pod tu hodnotou jako 0 (většinou je ta mez širší jen bod). Jedna taková hodnota je 1 bit (b). Přenášíme tzv. sériovým přenosem, každý bit chvíli (diagram zobrazující tenhle přenos je tzv. timing diagram).

- 1 a 0 jde i jinak, přes dva vodiče, podle směru jejich rozdílu (kladný vs. záporný rozdíl), což má navíc výhodu, že jsou šumem ovlivněny téměř stejně. Tzv. diferenciální přenos (např. USB).

*Pozor*

Napětí je relativní! (Musíme, krom diferenciálního přenosu, měřit oproti nějaké „nule“, tzv. zemi (ground).)

*Pozor*

Zdroj pracuje jen v uzavřeném obvodu!

*Poznámka* (Mocniny 2)

Hodí se naučit se mocniny 2:  $2^4 = 64$ ;  $2^8 = 256$ ;  $2^{10} = 1024$ ;  $2^{12} = 4096$ ;  $2^{16} = 65536$ ;  $2^{20} \approx 10^6$ ;  $2^{30} \approx 4 \cdot 10^9$

**Definice 3.1** (Most significant bit (MSb))

Bit s nejvyšší hodnotou (nejvyšší mocninou 2).

**Definice 3.2** (Least significant bit (LSb))

Bit s nejnižší hodnotou (nejnižší mocninou 2, typicky  $2^0$ ).

**Definice 3.3** (Bitorder)

Pořadí, v jakém se bity posílají (MSb-first / LSb-first)

**Definice 3.4** (Přenosová rychlost (transfer rate))

Rychlost, udává se v b / s (bps) nebo bauds (=symbols=včetně nedatových věcí jako mezery mezi „pakety“) / s.

**Definice 3.5** (Idle stav linky)

Když se nic nepřenáší (lze vytvořit stavem vysoké impedance, nebo tím, že definujeme 1 / 0 jako idle stav a přenos je započat tzv. start condition = změna na 0 / 1 (jelikož je lepší start condition mít delší, tak se vysílá délkou 1 bit tzv. startbit)).

**Definice 3.6** (stav Vysoké impedance = high Z)

Místo zapojení do 1 / do 0 vypojíme vysílací linku. Tento stav pak lze detekovat u přijímajícího).

*Poznámka (Hodiny a synchronizace)*

Start condition (začátek start condition je tzv. raising edge, na ní se právě synchronizuje) může dokonce synchronizovat čas.

Navíc hodiny nejsou přesné, takže po nějaké době budou rozsynchronizované. To lze vyřešit buď omezením délky přenosu. Většinou se však definuje délka, protože posílat end znak je těžké, tedy se dohodlo 8 bitů = 1 byte = 8b = 1B. (Avšak starší počítače měli i jinou velikost 1B).

Po 1 přenosu však musí být pauza, aby se mohli zase synchronizovat hodiny, tedy se přenáší stopbity (začínají falling edge) (musí se dohodnout jejich počet, často 1).

To už začíná být moc zbytečných bitů, proto se přidává další signál – hodinový (střídající se 1 a 0, na hranách (ať už vzestupných nebo sestupných podle dohody, výjimečně ddr = double data rate = obojí) je střed bitu). Hodinový signál může vycházet buď od odesílatele, nebo z vnějšího zdroje pro oba.

Ten má ale problém, že se může (třeba různými délkami drátu) rozladit. Proto existuje 3 řešení: pravidelná synchronizace na hranách přenášeného signálu (clock recovery = obnovování hodin), aby se to nerozblýjelo na dlouhých shodných úsecích, existuje zakódování 8 bitů do 10 (nebo jiné, podle dohody), aby se nestalo, že všech 8 (nyní 10) bude jen 1 nebo jen 0 (např. USB, nebo RS232).

*Poznámka (Duplexní přenos)*

Zatím jsme řešili jen jednostrannou komunikaci (= simplexní přenos), ale v mnoha reálných situacích chceme tzv. duplexní přenos.

Nejjednodušší je halfduplex = přenáší se oběma směry po jedné lince.

Fullduplex jsou naopak 2 nezávislé jednosměrné (simplexní) linky (mohou mít společný ground) (např. RS232). Např. v RS232 je navíc linka pro tzv. out of band signály (většinou buď 1 = pravda nebo 0 = nepravda, resp. tzv. inverzní logika značená overline) jako třeba warning nebo error (třeba 1 kabel pro informaci dochází baterka...).

### **Definice 3.7 (Řadič)**

Dále je nutné definovat komunikační protokol, který může definovat různorodé pakety. Přijímání (tzn. třeba přechod z 7b na 8b byte, nebo LSb-first / MSb-first) má na starost tzv. řadič (= controller), paměti v řadiči se říká registr (= register).

Registry jsou podle funkce pojmenovány např. datový, buffer, konfigurační (pro konfiguraci např. rychlosti linky...), ...