

- 1) Co je adresa v paměti? Jak taková adresa vypadá?
- 2) Co je adresový prostor paměti? Jak souvisí s kapacitou paměti?
- 3) Proč je 1 KiB velký právě 1024 bytů a nikoliv jen 1000 bytů, jak je běžné pro předponu kilo?
- 4) Pro následující kapacity pamětí napište (a) nejméně kolika bitový musí být adresový prostor, kterým jsme schopni celou takovou paměť uadresovat, (b) jaká by byla nejspíše typická velikost adresového prostoru (kolika bitový by byl), který by daná paměť používala ve svém komunikačním protokolu:
 - 128 B
 - 384 B
 - 2 kB
 - 8 kB
 - 64 kB
 - 128 kB
 - 1 MB
 - 2 GB
 - 500 GB
 - 4 TB
- 5) Paměť RAM je zkratkou za Random Access Memory – co se v tomto kontextu myslí slovním spojením „random access“ ... ? Jsou typické RAM paměti v tomto pohledu dnes opravdu „random access“ ... ?
- 6) Co určuje tzv. přenosová rychlost (transfer rate) paměti? Pro jaký druh přístupů k paměti je toto kritérium rozhodující?
- 7) Co určuje tzv. přístupová doba (access time) paměti? Pro jaký druh přístupů k paměti je toto kritérium rozhodující?
- 8) Srovnajte hlavní výhody a nevýhody pamětí SRAM vs. DRAM.
- 9) Co je tzv. slovo?
- 10) Co znamená, pokud o nějaké paměti řekneme, že je N-bitová? Souvisí N nějak s kapacitou paměti? A s její rychlostí?
- 11) Předpokládejte I²C SRAM paměť s níže uvedenými parametry kapacity a velikosti slova. Navrhněte komunikační protokol (tj. kompletní formát I²C paketu) pro zápis do této paměti (bez podpory burst přenosů). Čtení z paměti ve svém návrhu pro jednoduchost řešit nemusíte. Vyřešte pro všechny 4 varianty:
 - a) 32 KiB, 8-bitové slovo
 - b) 4 MiB, 8-bitové slovo
 - c) 4 MiB, 16-bitové slovo
 - d) 64 MiB, 32-bitové slovo
- 12) K čemu slouží tzv. adresový registr paměti? Co znamená jeho tzv. auto-increment?
- 13) Předpokládejte I²C SRAM paměť s níže uvedenými parametry, a pro ni běžný rozumný komunikační protokol bez podpory burst přenosů. Dále předpokládejte, že frekvence I²C sběrnice je 100 kHz. Pokud budeme do chtít do paměti zapsat N bytů, kde N je nějaký násobek 4:
 - a) na adresy 0, 8, 16, ..., (N-1)*8,
 - b) na adresy 0, 1, ..., N-1,tak spočítejte, kolik bude přibližně reálně dosažená přenosová rychlost (implikující délku přenosu těchto N bytů). Pro jednoduchost ignorujte délku start a stop condition a délku IDLE stavu. Vyřešte pro všechny 4 možné kombinace variant a), b), a níže uvedených variant x), y):
 - x) paměť používá 16-bitový adresový prostor, 8-bitové slovo,
 - y) paměť používá 16-bitový adresový prostor, 32-bitové slovo.Upozornění: Pro variantu y) si nezapomeňte promyslet, že v paměti chceme změnit jen výše uvedené byty a žádné jiné.

Sada programovacích úloh – společné poznámky: Vezměte si projekt PySimulatedI2c-SRAM-256B-ASSIGNMENT, který v souboru PySimulatedI2cMainProgram.py obsahuje kostru programu, který máte doplnit v řešení úloh 14, 15, 16. Součástí projektu je stejná implementace simulovaných příkazů CPU pro komunikaci po I²C sběrnici (resp. přesněji

simulovaného řadiče I²C sběrnice) jako v 5. sadě self-assessment úloh. Předpokládejte, že k simulované I²C sběrnici jsou současně připojená virtuální slave zařízení z 5. sady (která ale v úlohách této sady nebudeme používat) a navíc několik simulovaných SRAM pamětí typu *Philips PCF8570*. Při řešení úloh je třeba mít k dispozici datasheet tohoto slave zařízení (*Philips PCF8570*), viz přiložený soubor „06-PCF8570 - I2C serial SRAM memory.pdf“¹ (stejný jako v příloze 6. přednášky) – z datasheetu jsou pro nás relevantní pouze strany 3, 4, 7, 8, 11, 12. Na objektu `i2c` máte k dispozici 2 funkce stejně jako v sadě self-assessment úloh k 5. přednášce – v zadání 5. sady úloh také najdete detailní popis fungování těchto funkcí, a poznámky k jejich používání:

```
i2c.send(list-bytů-k-poslání-po-simulované-I2C-sběrnici)
i2c.receive(maximální-počet-bytů-k-přijetí-ze-simulované-I2C-sběrnice)
```

Poznámka: Řešení úloh 14, 15, 16 zkuste vypracovat „bez podvádění“, tj. pouze s využitím informací z přiloženého datasheetu, a bez nahlížení do souboru `SimulatedDevicePhilipsPCF8570_SRAM_256B.py`, který obsahuje implementaci simulace daného zařízení. V reálném životě by takové použité zařízení bylo pro vás také „černá skříňka“, a museli byste se řídit výhradně jeho datasheetem. Po dokončení a odladění vaší implementace SA úloh se samozřejmě můžete do uvedeného souboru podívat, a prostudovat si, jak je simulace naprogramována.

Poznámka: V řešení níže uvedených úloh pro jednoduchost neřešte a neošetřujte chybové stavy. Předpokládejte, že jsou všechna zařízení na I²C sběrnici správně připojena, a komunikace s nimi probíhá vždy bez chyb a dle specifikace. Pouze pokud již patříte mezi pokročilejší programátory, tak by mohlo být zajímavé ošetřit v programu i všechny okrajové situace, které by při komunikaci s reálným zařízením připojeným na reálné I²C sběrnici mohlo nastat.

14) Doplněte program v `PySimulatedI2cMainProgram.py` tak, aby provedl zápis 1 bytu a ověřovací čtení stejného bytu ze simulované SRAM paměti – více detailů viz TODO komentáře v části označené „exercise 14“.

15) Rozšiřte své řešení úlohy 14 o další operace s připojenou SRAM – více detailů viz TODO komentáře v části označené „exercise 15“.

16) Pro lehce pokročilé (je třeba si propojit souvislosti z 6. a 5. přednášky, a promyslet si, jak detekovat, že je nějaké zařízení připojené na I²C sběrnici): Doplněte program v `PySimulatedI2cMainProgram.py` tak, aby provedl operace uvedené v TODO komentářích v části označené „exercise 16“.
Pro řešení této úlohy nejsou nutná řešení úloh 14 a 15.

¹ Původní zdroj (listopad 2020): <https://www.nxp.com/docs/en/data-sheet/PCF8570.pdf>