

30. Procesory, FPGA obvody (realizace automatů pomocí procesorů a FPGA)

Procesor

- Komplexní **sekvenční** integrovaný obvod
- Obecně je složen z následujících částí:
 - řadič instrukcí
 - jedna nebo více aritmeticko-logických jednotek
 - speciální funkční registry (pomocné registry)
 - sběrnice propojující jednotlivé části
 - vstup a výstup (vyvedená sběrnice)
 - volitelně další podpůrné obvody, např:
 - matematický koprocessor (Floating Point Unit)
 - vyrovnávací paměť
- Každý procesor disponuje **instrukční sadou**, která obsahuje všechny elementární operace, které je procesor schopen vykonávat.
- Procesor ke své činnosti potřebuje zdroj hodinového signálu, ideálně krystal, který zajistí stabilní frekvenci. Každá instrukce je v procesoru vykonána během několika taktů (hodinových pulzů krystalu). Minimální počet taktů potřebných pro vykonání jedné instrukce odpovídá **instrukčnímu cyklu**.
- Dvě základní architektury procesorů:
 - **RISC Reduced Instruction Set Computer** – méně instrukcí, jednoduché instrukce, pevně daný instrukční cyklus (všechny instrukce trvají stejně dlouho).
 - **CISC Complex Instruction Set Computer** – více specializovaných instrukcí, nižší počet registrů, každá instrukce může zabrat různý počet instrukčních cyklů.
- K procesoru se obvykle připojuje paměť a řada dalších obvodů – různé řadiče periférií atd. Tím je tvořen celý automat, počítač, embedded systém apod. Tendence je taková, že se procesor zejména u dedikovaných zařízení stává součástí mikrokontroléru (nachází se ve stejném pouzdře). Mikrokontrolér již obsahuje všechny tyto přidané obvody, především **paměť a různá komunikační rozhraní**.
- Dvě základní architektury mikropočítačů:
 - **Harwardská** – oddělená paměť programu od paměti dat (zpravidla non-volatile EEPROM, FLASH pro program a volatilní SRAM, DRAM pro data), oddělené sběrnice pro instrukce, adresaci paměti a pro data
 - **Von Neumannova** – společná paměť pro data i program, společná sběrnice, nelze paralelizovat načítání instrukcí a zápis dat

- Základní parametry procesorů
 - Rychlost jádra (počet operací za vteřinu MIPS)
 - Šířka slova (sběrnice, operandu instrukcí)
 - Počet jader
 - Velikost vyrovnávací paměti,
 - Velikost adresovatelné paměti...

FPGA (Field-Programmable Gate Array)

- Speciální integrované obvody obsahující různé **univerzální programovatelné bloky a propojovací maticí**
- Kompromis mezi univerzálními součástkami jako jsou procesory nebo jednoduché číslicové obvody a tzv. **ASIC** (Application-Specific Integrated Curcuit). Obvody ASIC se navrhují přímo během výroby pro konkrétní nasazení.
- Obvody **FPGA** se dají nakonfigurovat přímo u zákazníka.
- Oproti procesorům mají výhodu, že lze vytvořit pomocí konfigurace nativní (hardwarovou) podporu pro danou aplikaci.
- Dva základní typy
 - FPGA s **volatilní konfigurací** – maticové propojky realizované pomocí buněk SRAM, snadná rekonfigurovatelnost za běhu systému, nutné nakonfigurovat po každém spuštění
 - FPGA s **non-volatilní konfigurací** – realizované pomocí EPROM, EEPROM, Flash, Fuse, Antifuse..., problematická rekonfigurace (u Fuse to ani není možné, pojistky v propojkách se přepálí a není návratu)..
- FPGA se neprogramují ve stejném smyslu jako procesory, popisují se popisovacími jazyky jako **VHDL** nebo **Verilog**. Při sestavování konfigurace se provede nejprve syntéza a pak implementace.
- V hradlových polích FPGA je možné realizovat skutečné procesory, pokud jsou tato pole dostatečně komplexní. Lze tedy do FPGA umístit jak procesor, tak ostatní aplikační logiku.

Možnosti realizace automatů

- Číslicové automaty lze realizovat mnoha způsoby, s ohledem na optimální poměr mezi cenou návrhu, realizace, dobou vývoje a požadavky na optimalizaci, efektivitu či výkon:
 - elementární číslicové součástky
Příliš velké, mnoho součástek, pro komplexnější aplikace nevhodné
 - ASIC
Drahý návrh a vývoj, maximální optimalizace, vhodné pro velké série
 - FPGA
Kompromis mezi ASIC a univerzálními součástkami
Možnost realizovat výkonově optimalizované aplikace za nižší náklady než při návrhu vlastních ASIC obvodů.
Pro vývoj aplikace se používají popisovací jazyky (VHDL, Verilog)
 - univerzální mikrokontrolér
Nutná SW realizace celé aplikace, cenově přijatelné, nižší optimalizace než ASIC nebo FPGA.
Snadné programování (C, Basic) eventuálně ASM
 - univerzální PLC (Programmable Logic Controller)
Omezené možnosti, nižší výkon, velká úroveň zapouzdření
Snadné zapojení a programování (Strukturovaný text atd.)