

# B35APO: Architektury počítačů

## Lekce 08. Výukový kit MZ\_APO (Xilinx Zynq MicroZed APO)

Pavel Píša

pisa@fel.cvut.cz

Petr Štěpán

stepan@fel.cvut.cz



19. dubna, 2023

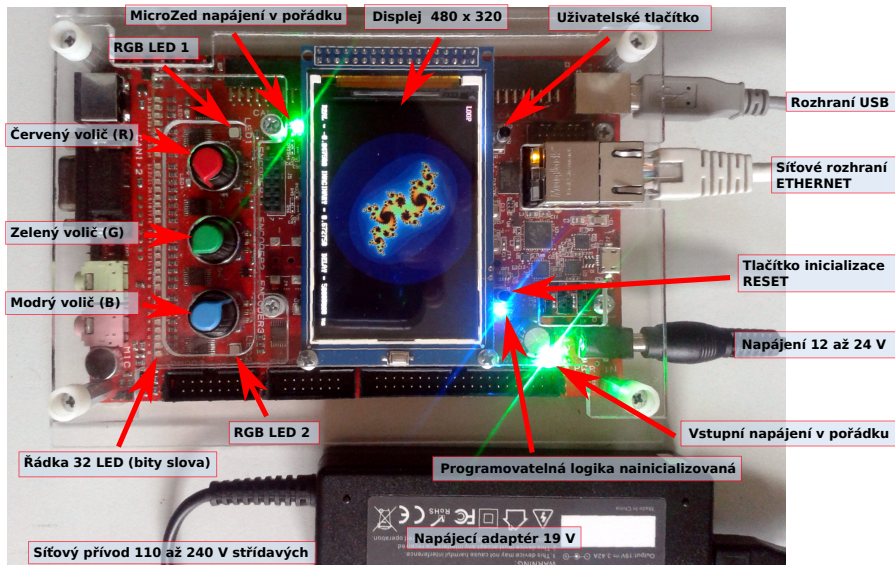
# Obsah

- 1** MZ\_APO – Xilinx Zynq MicroZed výukový kit pro B35APO
- 2 MZ\_APO – Periferie mapované do paměťového adresního prostoru

# Cíl dnešní přednášky

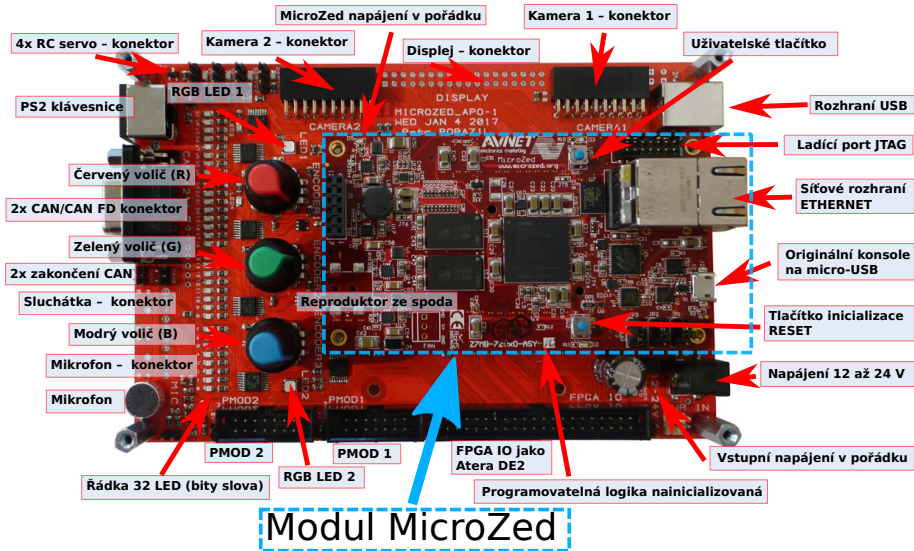
- Seznámení s komponentami/konstrukcí výukového kitu
- Komunikace a základní práce s kitem
- Princip a přístup na displej z tekutých krystalů (LCD)
- Barevné modely pro vykreslování
- Výstup písma
- Využití HW pro reálné aplikace

# Výukový kit MZ\_APO – smontovaný

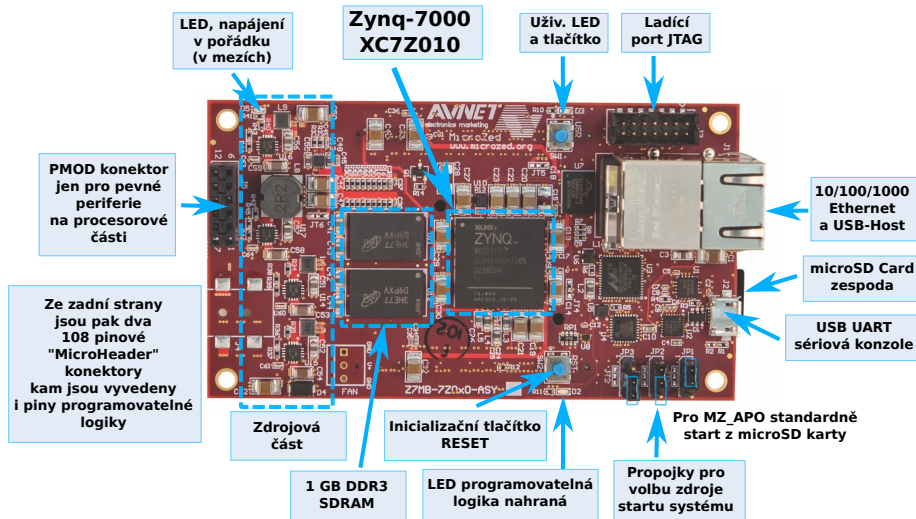




# Výukový kit MZ\_APO – základová deska



# Modul MicroZed – pohled shora

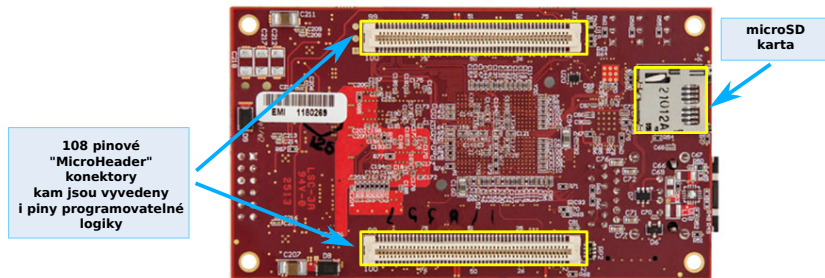


# Modul MicroZed – pohled zespodu

MicroZed Evaluation Kit – ADSAES-Z7MB-7Z010-G (případně AES-Z7MB-7Z010-SOM-G/REV-H cena 214 USD)

SoM – počítač na modulu (System on Module)

Čip Xilinx/AMD XC7Z010, cena okolo 90 USD (2023)



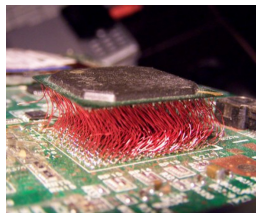
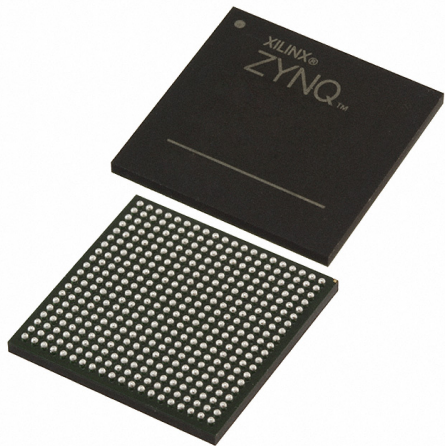
108-pinové konektory umožňují využít a propojit modul s vlastním návrhem. I přes tyto piny je možné modul napájet, přístup k PS periferiím i pinům programovatelné logiky PL (FPGA).

# Modul MicroZed – katalogový list

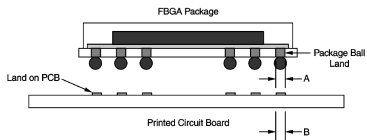
- Čip FPGA – Zynq™-7000 AP SoC (XC7Z010-CLG400-1)
  - CPU: Dual ARM® Cortex™-A9 MPCore™ @ 866 MHz
  - rychlá vnitřní statická paměť 256 kB
  - 4400 řezů (slice) - každý řez je malý konfigurovatelný logický obvod. Dokáže vytvořit až 8 klopných obvodů a 4 logické funkce se 6-ti vstupy. Uživatel je může libovolně konfigurovat a vzájemně propojovat. (28 K log. bloků, okolo 430 K ekviv. log. hradel)
  - 240 KB (60×36 kbit) RAM a 80×DSP (MAC)
- Externí dynamická paměť – 1 GB DDR3
- Komunikace – 10/100/1000 Ethernet
- MicroSD karta 4 GB. V desce APO obsahuje zavaděč systému Linux pro síť Ethernet.
- USB Host 2.0 a USB-UART
- Quad-SPI Flash 128 Mb pro inicializaci při zapnutí.
- V APO se nepoužívá.

# Zynq™-7000 AP – pouzdro FBGA

FBGA = Fine-Pitch Ball Grid Array



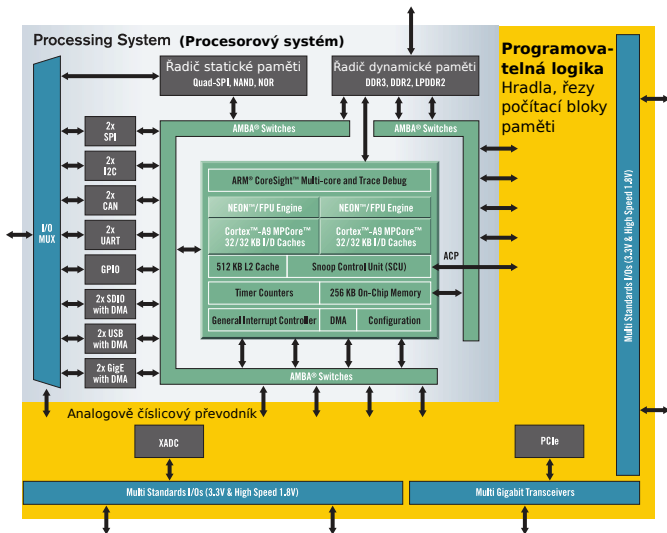
Kuriózní pájení



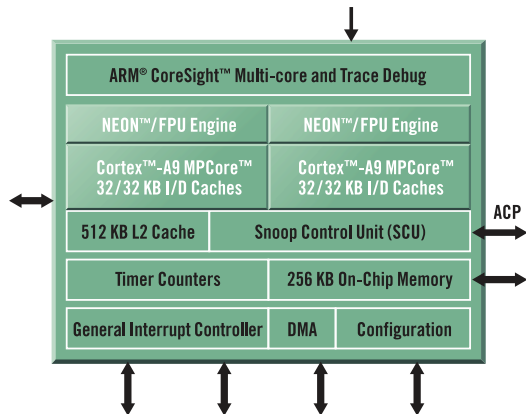
Korektní pájení



# Xilinx Zynq-7000 – Systém na čipu (SoC)



# Zvětšené jádro procesorové části



Již jsme v APO probrali

- 1 Processorové jádro
- 2 Aritmetiku v plovoucí řád. čárce (FPU)
- 3 Skrytou paměť (Cache) instrukční a datovou úroveň L1
- 4 Skrytou paměť další úrovně L2
- 5 Paměť na čipu (RAM)

Probereme

- 1 Koherenci paměti – sledovací jednotka (Snoop Control Unit)
- 2 Přímý přístup k paměti (Direct Memory Access)

# Příklady zařízení s Cortex-A9 jádrem

Seznam implementací [https://en.wikipedia.org/wiki/ARM\\_Cortex-A9#Implementations](https://en.wikipedia.org/wiki/ARM_Cortex-A9#Implementations)

Asus Transformer Pad, Infinity (TF700T)

[https:](https://en.wikipedia.org/wiki/Asus_Transformer_Pad_Infinity)

[//en.wikipedia.org/wiki/Asus\\_Transformer\\_Pad\\_Infinity](https://en.wikipedia.org/wiki/Asus_Transformer_Pad_Infinity)



Apple A5 (iPhone 4S, iPad 2, iPad mini)

[http://en.wikipedia.org/wiki/Iphone\\_4s](http://en.wikipedia.org/wiki/Iphone_4s)



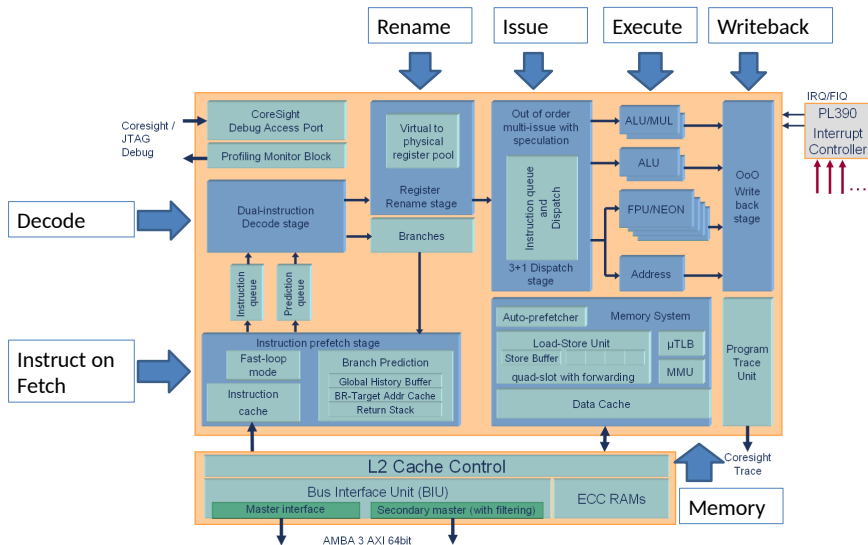
NVIDIA Tegra 2 (Motorola Xoom, Droid X2)

[http://en.wikipedia.org/wiki/Motorola\\_Xoom](http://en.wikipedia.org/wiki/Motorola_Xoom)





# Mikroarchitektura jádra Cortex-A9



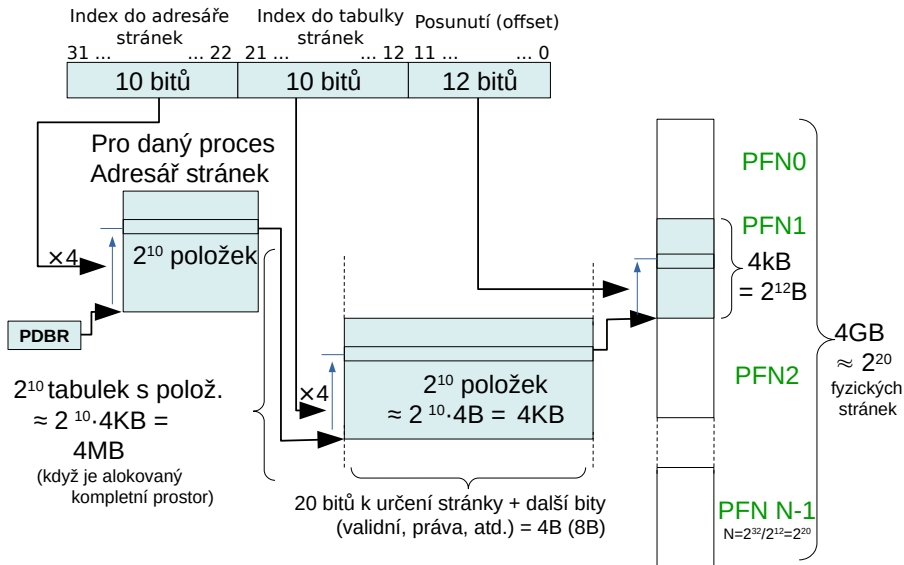
# Parmetry Zynq jádra Cortex-A9

- 32 bit RISC Little Endian, 16 celočíselných registrů
  - $2,5 \text{ DMIPS/MHz} \Rightarrow 866 \text{ MHz} * 2,5 \text{ DMIPS/MHz} = 2165 \text{ DMIPS}$   
Poznámka: DMIPS je výsledkem syntetického výpočetního benchmarku Dhrystone určeného k reprezentaci celočíselného programování
- Většina celočíselných instrukcí s latencí 1 cyklus, celočíselné násobení potřebuje 4 až 5 cyklů.
- Plovoucí řádová čárka prochází ALU 4 cykly pro sčítání, odečítání (FADD, FSUB), 5 cyklů pro násobení FMUL, 15 cyklů dělení FDIV (3x delší než násobení!), 17 cyklů druhé odmocniny FSQRT.
- Predikce větvení
  - Tabulka 4K záznamů 2-bitových prediktorů.
- Virtuální paměť se 2 úrovněmi stránkovacích tabulek

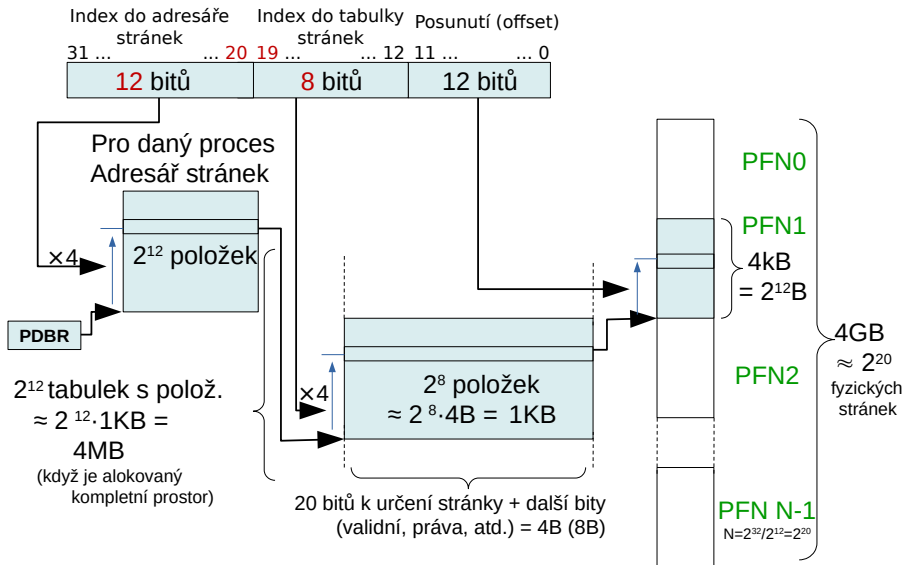
# Skrytá paměť úrovně jedna a dvě (anglicky Cache)

- 2 samostatné L1 SP pro instrukce I-cache a D-cache pro data. Vlastnosti pro obě L1:
  - velikost 32 kB
  - 4-cestně asociativní (4-way set associative)
  - délka bloku (řádky) 32 bajtů
  - politika nahrazování pseudo-náhodné nebo pseudo round-robin
  - D-Cache podporuje zpětný zápisu nebo alokace při zápisu (write-back/write-allocate policy).
- L2 SP je sdílena oběma jádry Cortex-A9. Vlastnosti:
  - velikost 512 kB
  - 8-cestně asociativní
  - délka bloku (řádky) 32 bajtů
  - politika výměny je pseudonáhodná,
  - podporuje zpětný zápis (write-back), přímý zápis (write-through) s a bez alokace

## Stránkování pro dvě úrovně, připomenutí z před. 4. pro x86



## Stránkování ARM Cortex-A9 – dvě úrovně ale nesymetrické

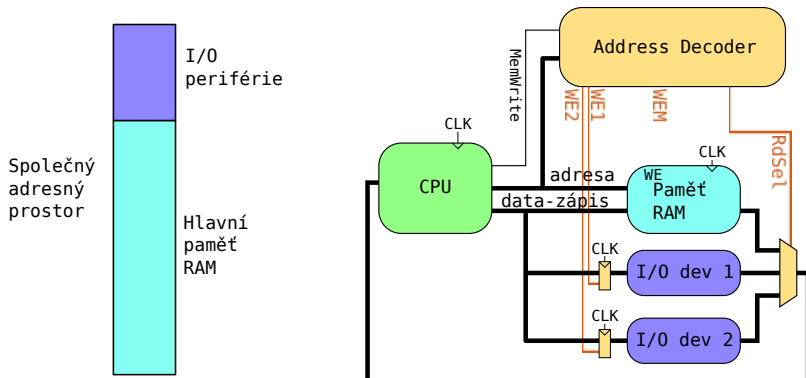


# Obsah

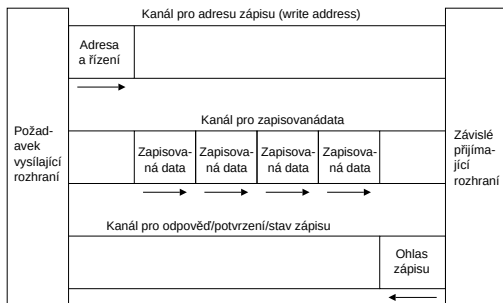
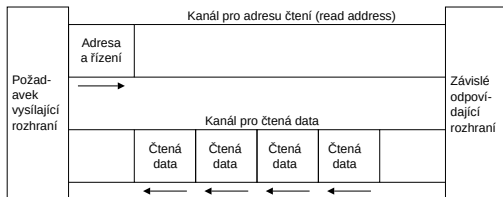
- 1 MZ\_APO – Xilinx Zynq MicroZed výukový kit pro B35APO
- 2 MZ\_APO – Periferie mapované do paměťového adresního prostoru

# Paměťově mapované periférie, připomenutí

- ARM také nemá speciální instrukce pro komunikace s perifériemi
- pro komunikaci s perifériemi se využívá ukládání a čtení z paměti
- Address Decoder – rozhoduje, kam se data přesměrují

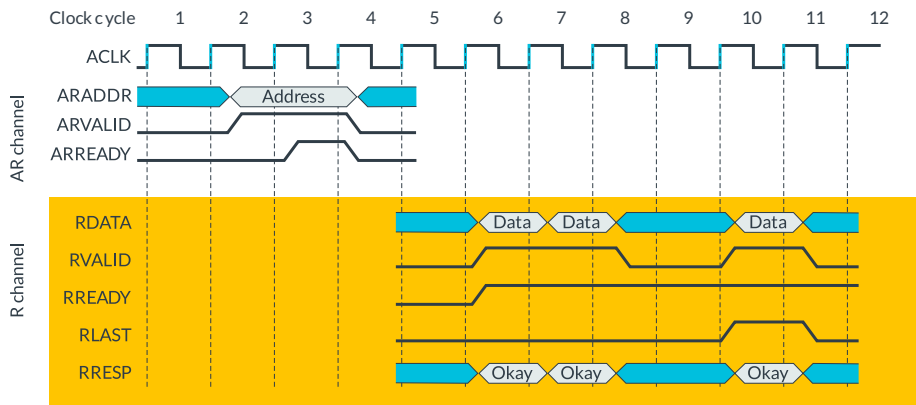


# Sběrnice AMBA AXI – princip kanálů



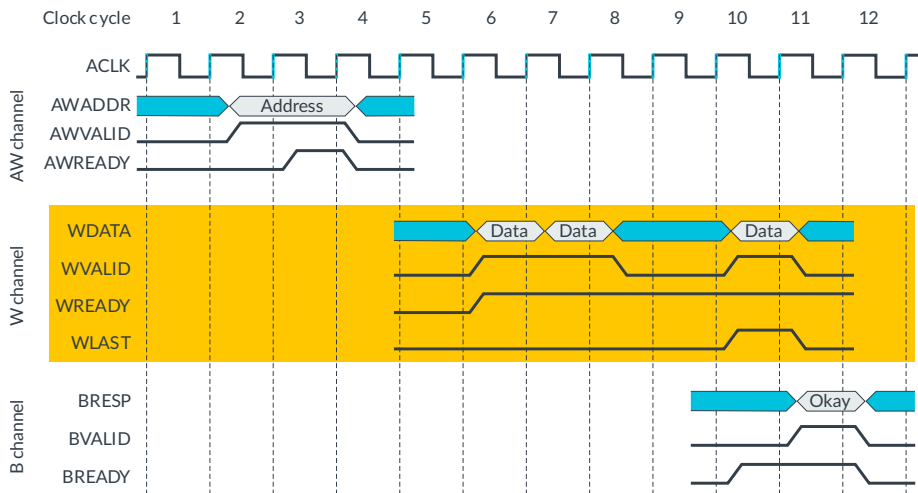


# Sběrnice AMBA AXI – čtení



- Samostatné kanály pro adresu a data
- Přenos proběhne vždy při soběhu xVALID a xREADY

# Sběrnice AMBA AXI – zápis



## Logický návrh MZ\_APO v programu Vivado

The screenshot displays the Vivado IDE interface for the MZ\_APO project. The main workspace shows a block diagram of the design, featuring a ZYNQ Processing System (processing\_system7\_0) connected to various peripheral blocks. Key components include:

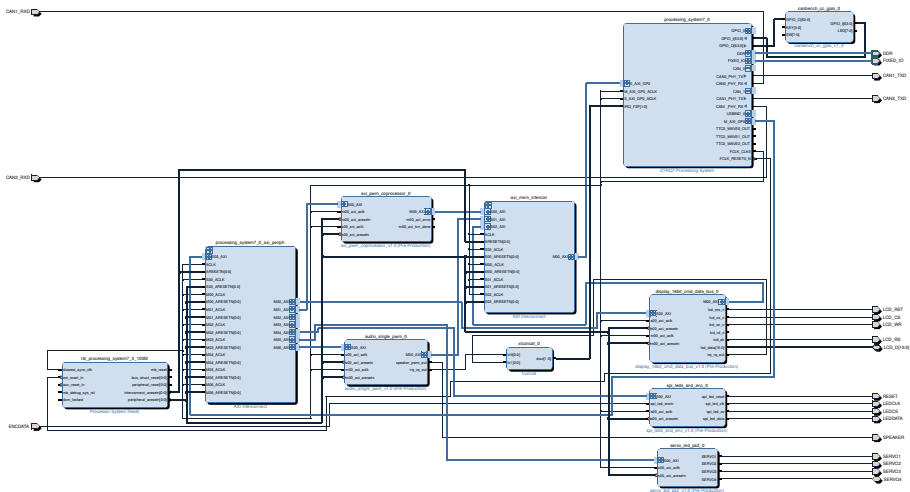
- axi\_mem\_intercon**: An AXI Interconnect block with external ports for `MEM_AR`, `MEM_AW`, `MEM_AR_C2S`, `MEM_AW_C2S`, `MEM_AR_C2M`, `MEM_AW_C2M`, `MEM_AR_M2S`, `MEM_AW_M2S`, `MEM_AR_M2M`, and `MEM_AW_M2M`.
- display\_16bit\_cmd\_data\_bus\_0**: A block with ports for `MEM_AR`, `MEM_AW`, `MEM_AR_C2S`, `MEM_AW_C2S`, `MEM_AR_C2M`, `MEM_AW_C2M`, `MEM_AR_M2S`, `MEM_AW_M2S`, `MEM_AR_M2M`, and `MEM_AW_M2M`.
- carbench\_cc\_apio\_0**: A peripheral block with ports for `GPO_CLK0`, `GPO_CLK1`, `MEM_AR`, `MEM_AW`, `MEM_AR_C2S`, `MEM_AW_C2S`, `MEM_AR_C2M`, `MEM_AW_C2M`, `MEM_AR_M2S`, `MEM_AW_M2S`, `MEM_AR_M2M`, and `MEM_AW_M2M`.

The left sidebar shows the Project Manager and Hierarchy. The bottom panel displays the Tcl Console with the following commands and output:

```

Adding cell -- xilinx.com:ip:axi_protocol_converter:2.1 - auto_pc
Adding cell -- xilinx.com:ip:axi_protocol_converter:2.1 - auto_pc
Adding cell -- xilinx.com:ip:axi_protocol_converter:2.1 - auto_pc
Adding cell -- xilinx.com:ip:axi_protocol_converter:2.1 - auto_pc
Successfully read diagram <top from 00 file = /home/pi/.xilinx/vivado/canbench-sw/system/src/top/top.bd>
open_bd_design Time (s): cpu = 00:00:24 ; elapsed = 00:00:19 - Memory (MB): peak = 6008.051 ; gain = 153.621 ; free physical = 80 ; free virtual = 7898
set_property location {-22 4633} [get_bd_ports CAN2_N0]
set_property location {-20 4136} [get_bd_ports CAN2_DATA]
write_bd_layout -format pdf -orientation portrait /home/pi/mz_apo-v10-top.pdf
./home/pi/mz_apo-v10-top.pdf
  
```

## MZ\_APO – logický návrh, propjení sběrnic a bloků



# MZ\_APO – Fyzické adresy výukových periférií

Bázové adresa	Délka	určení
0x0000 0000	1 GB	DRAM
0x4000 0000	1 GB	port 0 sběrnice AXI do programovatelné části
<b>0x43c0 0000</b>	16 bytů	<b>APO – LCD displej</b>
0x43c2 0000	32 bytů	stejnsměrný motor na PMOD1
0x43c3 0000	32 bytů	stejnsměrný motor PMOD2
<b>0x43c4 0000</b>	48 bytů	<b>APO – periferie voličů, indikátorů</b>
0x43c5 0000	32 bytů	4× Servo a PS2 klávesnice
0x43c6 0000	32 bytů	jednoduché audio
0x8000 0000	1 GB	port 1 sběrnice AXI do programovatelné části
0xE000 0000		Reservované pro systém
0xFFFC 0000		256 kB interní RAM

# Mapování rozsahu fyzického adresního prostoru do virtuálního

```

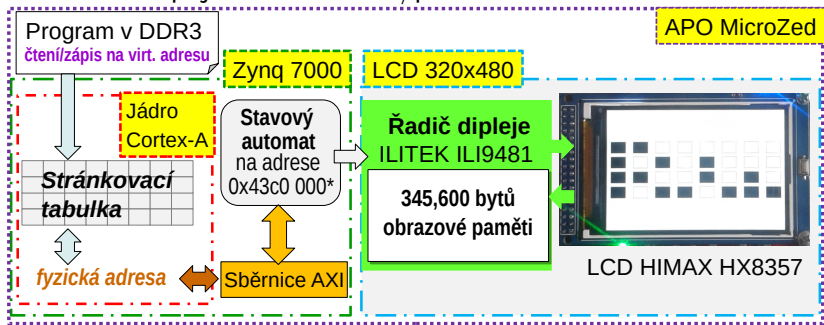
int fd = open("/dev/mem", /* reprezentuje fyzickou paměť, cíl přístup */
             O_RDWR /* s oprávněním pro čtení i zápis */
             | O_SYNC /* a bez použití skryté paměti pro /dev/mem */
             );
unsigned char *mem = (unsigned char *) mmap(
    NULL, /* virtuální adresa nspecifikovaná, vybere jí jádro */
    0x4000 /* požadovaný rozsah mapování do virtuálního prostoru */,
    PROT_READ | PROT_WRITE, /* s přístupem pro čtení a zápis */
    MAP_SHARED, /* změny viditelné ostatním procesům a perif. */
    fd, /* rukověť k již otevřenému souboru (file handle, descriptor) */
    0x43c40000 /* posun v rámci souboru, zde fyzická adresa I/O */ );

```



# Zápis na displej na kitu MZ\_APO

Konečný automat (FSM) přenáší data na LCD displej generováním příslušných signálů pro řídicí čip, který periodicky obnovuje LCD TFT displej. Pokud se objeví nový požadavek na automat dříve, než je hotový předchozí přenos, automat zbrzdí AXI sběrnici negací READY, dokud nebude moci přijmout další data/příkaz.





# Zápis na displej na kitu MZ\_APO

Posun	Datový typ	Příkaz pro čip
+0x0	uint16_t	0x1 - inicializace/reset displeje, bit0 == 0 - vypnutí
+0x8	uint16_t	řídící příkaz pro kontrolér, 0x2c body od začátku
+0xC	uint16_t	zápis 16 bitové barvy bodu (RGB565) nebo jiných data
+0xC	uint32_t	zápis 2 po sobě následujících bodů: bity 15..0 a pak bity 31..0

