

Systemy wbudowane - wykład do pracy własnej komponenty systemów wbudowanych

Przemek Błaśkiewicz

22 kwietnia 2020

Układy w systemach wbudowanych

Składniki (każdego) systemu (*in order of appearance*):

- jednostki obliczeniowe (CPU) – zarządzają całością (mogą być małe) *mikrokontrolery, CPLD, FPGA, mikroprocesory* ;
- jednostki pomocnicze (ko-procesory) – realizują konkretne funkcjonalności, wyspecyfikowane *koprocesor krypto, układ FFT, kodek audio*;
- pamięci – stałe (np. flash), ulotne (RAM) – przechowują program, dane;
- przetworniki – np. ADC, układy radiowe, sensory zjawisk fizycznych *termometr, chipset Bluetooth, czujnik wstrząsu*
- kontrolery we/wy (I/O) – układy komunikacyjne, sterujące *układy I2C, UART, kontroler ekranu*
- kontrolery magistrali, mostki (bridge) – realizują połączenia podzespołów.

Mogą być to układy oddzielne (np. oddzielne kości scalone), lub złączone w jednej obudowie (SoC – *System-on-Chip*), wtedy dochodzą do tego układy zarządzania energią, często też inne peryferia: JTAG ♠, wewnętrzne zegary, ...).

(Główne) jednostki obliczeniowe

Wprowadźmy sobie podział ze względu na:

- Funkcjonalność procesora:
 - Ogólnego przeznaczenia: intele, AMD, ARMy... Potrafią dużo (wszystko?) ale realizacja niektórych zadań wymaga specjalnego podejścia, dodatkowych układów etc.
 - Programowalno-dedykowany, np. mikrokontrolery dedykowane są do ... kontrolowania (mają dużo sterowalnych pinów we/wy, dodatkowo wbudowane układy np. ADC, UART... – ich ideą jest „bycie sercem” układu sterowania)
 - Specjalizowany (custom, single-operation); robi jedną rzecz, ale dobrze. A innych nie robi, albo baaaardzo źle.
- Technologia wykonania układu:
 - CMOS – układy scalone, krzemowe, ogólnodostępne.
 - FPGA – układy z modyfikowalną strukturą (zob. poprzedni wykład). Można je dostosować do potrzeb, ale programowanie jest trudniejsze.
 - CPLD – układy programowalne, modyfikowalne, nieco lepiej przystosowane do zadań sterowania (więcej bloków we/wy niż typowy FPGA).

Procesor ogólnego przeznaczenia

Charakterystyka:

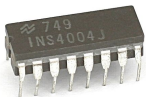
- uniwersalny procesor o szerokim zakresie *typowych* operacji;
- zazwyczaj ♠ architektura von Neumana (pamięć programu, pamięć danych);
- w oparciu o rejestry i dużą pamięć (zatem cache, zatem zarządca pamięci);
- potokowość, przepustowość, szeroka magistrala → szybkość działania;

Zalety:

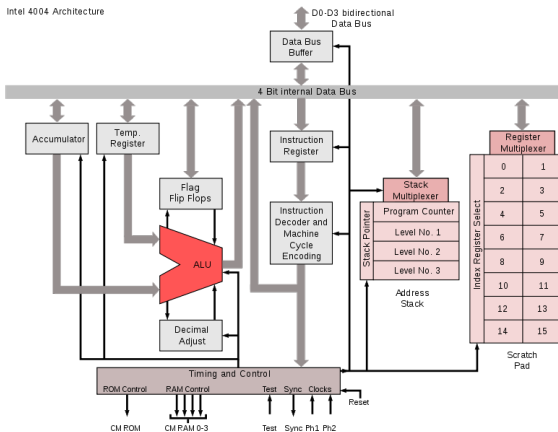
- szybka produkcja (łatwiejsze programowanie);
- łatwa dostępność;
- wysoka elastyczność - jedna aplikacja, wiele możliwości.

Wadą jest **koszt jednostkowy**. 30\$ za procesor do sterowania nawadnianiem to trochę za dużo. No i do tego są potrzebne peryferia.

Intel 4004 (1971 r.)



- 4-bit dane
- 8-bit instr.
- 12-bit adr.
- zegar 740kHz
- 2300 tranzystorów
- 8 cykli/instr.
- arch. harwardzka ♠



♠ Porównaj schemat blokowy 4004 ze schematami TMS320, rodziny AVR i HD44780, które pojawią się w tej prezentacji.

Pewne operacje są typowe dla pewnych dziedzin/zastosowań. I tak np. DSP opiera się na modyfikowaniu sekwencji próbek sygnałów, w tym *filtrowaniu*.

Filtry (grafika, audio, radio cyfrowe) to często obliczenie:

$$y = \sum_{k=0}^{M-1} A_k X_k$$

A_i to parametry filtra, X_k to próbki wejściowe, y to *odpowiedź* filtra.

Przyjrzyjmy się tej operacji, zwanej MAC (*multiply and add*). Niech próbki wejściowe będą N -bitowe, wtedy m -ta próbka to $X_m = b_{m(N-1)}b_{m(N-2)} \dots b_m 1 b_m 0$. Wtedy obliczenie MAC można rozpisać:

$$\begin{aligned} y &= \sum_{k=0}^{M-1} A_k \left[b_{k(N-1)} 2^{N-1} + \sum_{n=0}^{N-2} b_{kn} 2^n \right] \\ &= \sum_{k=0}^{M-1} A_k b_{k(N-1)} 2^{N-1} + \sum_{n=0}^{N-2} \left[\sum_{k=0}^{M-1} A_k b_{kn} \right] 2^n \end{aligned}$$

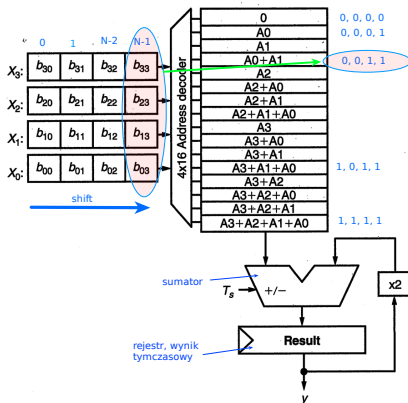
Procesor programowalno-dedykowany

$$y = \sum_{k=0}^{M-1} A_k b_{k(N-1)} 2^{N-1} + \sum_{n=0}^{N-2} \left[\sum_{k=0}^{M-1} A_k b_{kn} \right] 2^n$$

Wartości sum częściowych

$\sum A_k b_{kn}$ tabularyzujemy dla wszystkich możliwych b_{kn} .

Pierwszy składnik (dla $N - 1$) jest tylko przeskalowaną taką sumą (a więc shift bitowy). Drugi składnik to suma odpowiednich sum częściowych mnożonych cyklicznie przez 2. Wyboru sumy częściowej dokonujemy na podstawie 4 bitów z 4 próbek $X_{\{0,1,2,3\}}$, podawanych na dekoder 4-na-16.



Wyliczenie odpowiedzi filtra w N prostych krokach + preprocessing!

Procesory dedykowane, np. DSP (*digital signal processing*), czyli pewnie powtórka z AKiSO:

- przyspieszone operacje (np. MAC, wektorowe, FFT) – dedykowane układy mnożące oraz specjalne wywołania w asemblerze (mikropolecenia); specjalne tryby adresowania (np. „oblicz odpowiedź filtra o parametrach z adresu A na obszarze pamięci od B do C”).
- architektura harwardzka (dane programu można pobierać niezależnie od dekodowania instrukcji) i potokowość – idealnie pasuje do wykonywania tych samych operacji na blokach pamięci.
- trudniej programować (potrzebna większa wiedza, mniej specjalistów), wolniejsze w ogólnych zastosowaniach...

TMS320C5504/5 Block Diagram and Deltas

C5504 Features

■ Core

- Dual MAC, C55x CPU with JTAG disable option
- 1.05V @ 60MHz, 1.3V @ 100MHz (120MHz available)
- Dynamic Voltage and Frequency Scaling

■ Memory

- 256-KB On-Chip Memory: 64-KB DARAM, 192-KB SARAM
- 128-KB ROM

■ Peripherals

- Four serial busses offering combinations of PS, UART, SPI, MMC/SD, and GPIO
- High speed USB 2.0
- Multi-master and Slave PC with 7 or 10-bit addressing modes
- Three 32-bit timers with watchdog functionality
- Four 4-Channel DMAs
- 16-bit EMIF with asynchronous SRAM, NAND (with 4-bit ECC)
- Low power PLL (0.7mA) with 32KHz crystal oscillator
- Real-time clock with 32-KHz crystal input, 1 analog LDO

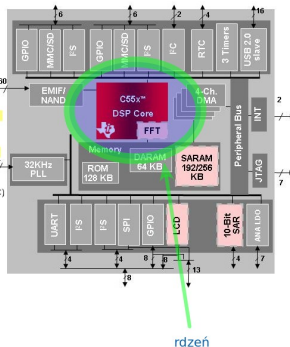
■ I/O

- 1.8V, 2.5V, 2.8V, 3.3V
- Separate I/O supplies for EMIF and serial interface

■ Package: 196-pin 10x10mm BGA with 0.65mm pitch

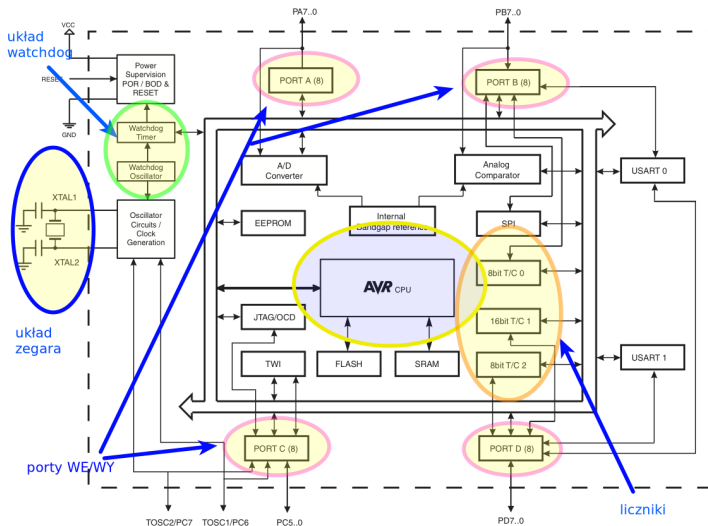
C5505 Additional Features:

- 320-KB On-Chip Memory: 64-KB DARAM, 256-KB SARAM
- 1024-point FFT Coprocessor
- LCD Bridge muxed with PS, SPI, UART, and GPIO
- 4ch 10-bit SAR ADC



Układy mikrokontrolerów często budowane są wokół rdzenia opracowanego przez danego producenta. Peryferie stanowią o rodzaju/przeznaczeniu układu.

Mikrokontrolery AVR – ATtiny, ATmega, X-Mega...



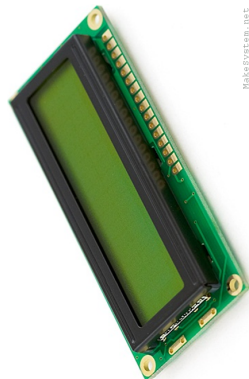
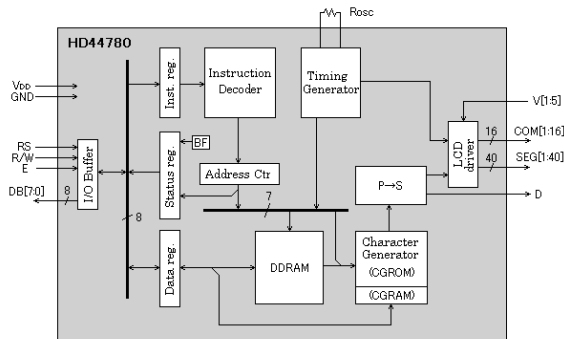
♠ Sprawdź, czy potrafisz określić, do czego służą komponenty zaznaczone na diagramie.

Mały układ wykonujący jedną, (prostą) czynność.

- timer/watchdog/counter;
- UART/USART
- kontroler LCD, silnika krokowego...

Są to gotowe procesory lub budowane własnym sumptem z jeszcze mniejszych komponentów...

HD44780 - kontroler LCD 2x16 znaków

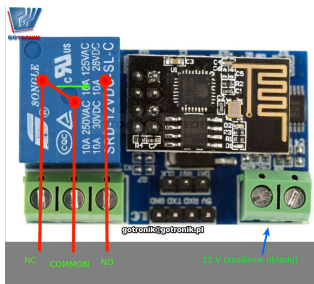
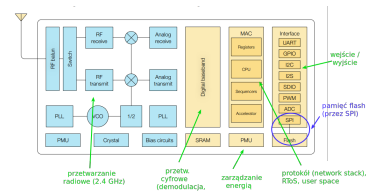


Bardzo powszechnie stosowany układ do zarządzania wyświetlaczami. Łączy się z mikrokontrolerem za pomocą kilku linii danych (RS, R/W, E oraz 4 lub 8 bitów DB), a wystawia 40 linii segmentu i 16 linii instrukcji w module wyświetlacza.

Posiada własną pamięć znaków (można definiować swoje), wewnętrzny układ taktowania, można go „programować”.

♠ znajdź notę katalogową i sprawdź, jakie komendy są dostępne.

Układ realizujący łączność radiową (2.4 GHz: WiFi, BT) oraz dający użytkownikowi możliwość programowania wbudowanego procesora. Możliwość sterowania pinami WE/WY. Dostępne SDK.



Sterowanie przez WiFi: układ ESP8266 (czarny) udostępnia łączność i steruje jednym pinem, który przełącza przekaźnik (niebieski). NC (normally closed) – połączone z COMMON, gdy nie ma zasilania, NO (normally open) – łączy się z COMMON, gdy przekaźnik jest włączony. ~30 zł w Gotronik ;)

Dalsze slajdy dla zainteresowanych...

COordinate Rotation DIgital Computer

Zestaw parametryzowanych algorytmów aproksymujących różne funkcje matematyczne (sin, cos, cosh, atan, pierwiastek...)

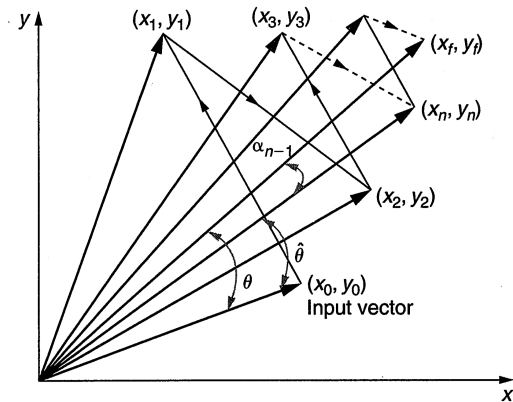
- operacje dodawania i przesunięcia, prosta arytmetyka
- efektywny dla wieloelementowych wektorów we/wy

CORDIC – obrót wektora (x_s, y_s) o θ

- $$\begin{aligned}x_f &= x_s \cos \theta - y_s \sin \theta \\y_f &= x_s \sin \theta + y_s \cos \theta\end{aligned}$$
- $$\theta = \sum_{i=0}^{\infty} \alpha_i$$
- $$\begin{aligned}x_{i+1} &= x_i \cos \alpha_i - y_i \sin \alpha_i \\y_{i+1} &= y_i \sin \alpha_i + x_i \cos \alpha_i\end{aligned}$$
- $$\begin{aligned}x_{i+1} &= \cos(\alpha_i)(x_i - y_i \tan(\alpha_i)) \\y_{i+1} &= \cos(\alpha_i)(y_i + x_i \tan(\alpha_i))\end{aligned}$$
- $$\alpha_i = \tan^{-1}(\sigma_i 2^{-i})$$
- $$\begin{aligned}x_{i+1} &= \cos(\alpha_i)(x_i - \sigma_i y_i 2^{-i}) \\y_{i+1} &= \cos(\alpha_i)(y_i + \sigma_i x_i 2^{-i})\end{aligned}$$

$$\begin{aligned}i &= 0 \\x_0 &= x_s \\y_0 &= y_s \\z_0 &= \theta \\x_{i+1} &= x_i - \sigma_i y_i 2^{-i} \\y_{i+1} &= y_i + \sigma_i x_i 2^{-i} \\z_{i+1} &= z_i - \sigma_i \tan^{-1}(2^{-i}) \\\sigma_i &= \begin{cases} 1 & \text{if } z_i \geq 0 \\ -1 & \text{if } z_i < 0 \end{cases}\end{aligned}$$

CORDIC



CORDIC

