

Laboratorium Mikroinformatyki
Emulatory Sprzętowe

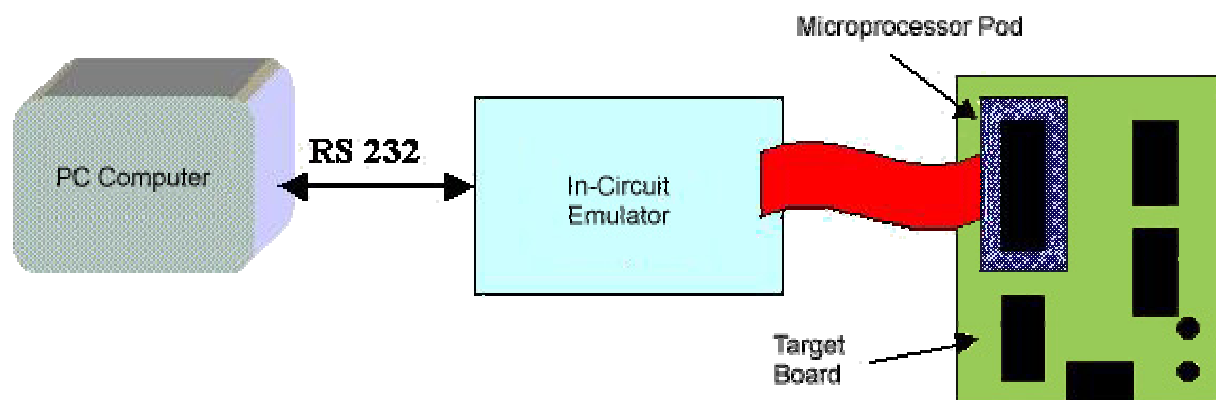
Co to jest Emulator ?

Emulator sprzętowy (In Circuit Emulator - ICE) jest nieocenionym narzędziem wspomagającym programistę przy tworzeniu oprogramowania dla systemów mikroprocesorowych. Mikroprocesor lub mikrokomputer jednoukładowy zastosowany w docelowym urządzeniu jest zastępowany przez emulator. Najczęściej w układ włączana jest sonda sterowana przez zasadniczą część emulatora za pośrednictwem wielożyłowego przewodu.

Emulator może zastępować pracujący mikroprocesor w czasie rzeczywistym. Programista ładuje program wykonywalny do emulatora, który umożliwia mu wykonywanie programu w całości, zatrzymywanie go w dowolnym momencie, wykonywanie programu krokowo lub zastawianie pułapek.

Większość emulatorów posiada wiele zaawansowanych funkcji jak analiza szybkości wykonywania programu, analiza przebiegu ścieżek programu, analiza danych, zaawansowane możliwości definicji warunków zatrzymywania programu (pułapki).

W skład sondy lub samego emulatora wchodzi procesor, który jest emulowany. Często jest to specjalna wersja takiego układu z wyprowadzonymi na zewnątrz wewnętrznymi sygnałami sterującymi i magistralami. Umożliwia on pełną analizę zawartości rejestrów mikroprocesora.



Emulator jest jednym z wielu narzędzi uruchomieniowych, jest jednak narzędziem najbardziej uniwersalnym.

Systemy mikrokomputerowe są trudniejsze w uruchamianiu niż programy uruchamiane na komputerach osobistych. W większości przypadków nie posiadają wygodnego i przejrzystego wyświetlacza i wielostykowej klawiatury co utrudnia wprowadzanie informacji oraz przeglądanie stanu rejestrów i pamięci urządzenia. Emulator jest jednym z najstarszych narzędzi uruchomieniowych dla takich systemów. Jest jedynym urządzeniem, które zastępuje mikroprocesor własnym i dzięki temu daje projektantowi możliwość pełnej kontroli zarówno nad tworzonym oprogramowaniem jak i nad pozostałą sprzętową częścią projektu. Za pomocą wyrafinowanych technik monitoruje wszystko co dzieje się w układzie dając pełny wgląd w działanie poszczególnych operacji. Jest mostem pomiędzy docelowym urządzeniem a komputerem osobistym, na którym tworzone jest oprogramowanie.

Wybrane funkcje emulatora.

Dostęp do sprzętu.

Najważniejszą funkcją emulatora jest pełny dostęp do urządzenia docelowego: możliwość sprawdzania i zmiany zawartości rejestrów, pamięci oraz stanów wejść i wyjść. Pozwala to na testowanie wadliwie działającego oprogramowania, ale przede wszystkim na wykrywanie błędów w projekcie sprzętu. Można na przykład, po napisaniu odpowiedniego oprogramowania, wykrywać usterki w układzie elektronicznym za pomocą oscyloskopu.

Punkty zatrzymania.

Punkty zatrzymania (breakpoint) to kolejna ważna funkcja emulatorów. Dają możliwość wstrzymania wykonywania programu w wybranym jego miejscu lub przy wystąpieniu określonych warunków. Zatrzymywanie może również nastąpić po każdej wykonanej instrukcji asemblerowej lub po grupie instrukcji tworzących linię programu w języku wysokiego poziomu.

Istnieje znacząca różnica pomiędzy dwoma typami punktów zatrzymania używanych przez programy uruchomieniowe. Programowe punkty zatrzymania działają na zasadzie zastąpienia wybranej instrukcji instrukcją przerwania. Jest niemożliwe stosowanie takich punktów w programie zawartym w pamięci stałej ROM. Emulatory umożliwiają ustalanie sprzętowych punktów zatrzymania, które sprawdzają warunki wstrzymywania wykonywania programu przed każdą wykonywaną instrukcją. Sprzętowe punkty zatrzymania działają zarówno dla programów zapisanych w pamięci RAM jak i ROM, nawet dla nieużywanych zakresów przestrzeni adresowej mikroprocesora.

Pamięć programu.

Program wynikowy jest zwykle przechowywany w pamięci stałej typu ROM, EEPROM lub Flash. Podczas uruchamiania zwykle istnieje potrzeba wielokrotnego przeprogramowania tej pamięci coraz to nową wersją oprogramowania. Emulatory przechowują program w wewnętrznej, szybkiej pamięci RAM, która jest logicznie przemapowywana w miejsce pamięci programu mikroprocesora pozwalając na wielokrotne szybkie przeładowanie kodu programu.

Ochrona pamięci.

Wiele emulatorów jest wyposażonych w mechanizmy ochrony pamięci przed nieoczekiwanymi odwołaniami. Program jest zatrzymywany kiedy przykładowo podejmie próbę zapisu do niedozwolonego obszaru pamięci w tym również do części pamięci zawierającej program.

Symulacja.

Za pomocą emulatora można uruchamiać oprogramowanie bez działającej części sprzętowej urządzenia. W takim przypadku emulator zachowuje się podobnie jak symulator wybranego mikroprocesora.

Śledzenie.

Śledzenie (trace) w czasie rzeczywistym jest jedną z ważniejszych opcji emulatorów, dostępną praktycznie wyłącznie przy ich użyciu. Śledzenie zapamiętuje wyniki działającego z pełną prędkością programu w buforze. Zachowuje stan nawet kilkuset tysięcy cykli mikroprocesora zapamiętując stan wybranych rejestrów, wystawiane adresy, pobierane instrukcje i przesyłane dane. Oprogramowanie emulatora pozwala na przetłumaczenie

zawartości bufora na program assemblerowy lub nawet na program w języku wysokiego poziomu ułatwiają jego analizę.

Mechanizmy wyzwalania.

Śledzenie jest zawsze powiązane z zaawansowanymi mechanizmami wyzwalania (trigger). W zależności od wyników działania programu można ustalić warunki rozpoczęcia i zakończenia śledzenia. Przykładowo można zapamiętać i prześledzić działanie rzadko wywoływanej procedury obsługi przerwania.

Problemy występujące w praktyce.

Problemy jakie mogą pojawić się przy używaniu emulatorów podczas uruchamiania urządzeń związane są z szybką ewolucją mikroprocesorów. Producenci prześcigają się w miniaturyzacji układów scalonych i zwiększaniu prędkości ich działania. Małe układy wymagają precyzyjnych sond i sprawiają trudności w połączeniu zwłaszcza przy montażu powierzchniowym SMD. Szybkości mikroprocesorów sięgające setek MHz również utrudniają pracę emulatorów. Aby w pełni emulować pracę mikroprocesora w czasie rzeczywistym muszą nadążyć z prędkością działania. Przszkodą są zwykle przewody łączące sondę z główną częścią emulatora, które przy dużych częstotliwościach pracy zachowują się jak linia długa.

Pewnym utrudnieniem może być również mnogość opcji i funkcji dostępnych w oprogramowaniu obsługującym emulator. Wielu użytkowników potrzebuje jedynie podstawowych opcji. Bardziej zaawansowane funkcje pułapek, śledzenia i wyzwalania potrzebne są wyłącznie do uruchamiania bardzo złożonych projektów.

Opracowane na podstawie *Introduction to In-Circuit Emulators* Jack Ganssle, Embedded Systems Programming.

Cel ćwiczenia laboratoryjnego.

W laboratorium są dostępne dwa stanowiska wyposażone w emulatory mikrokomputerów jednoukładowych serii 8051. Jeden emulator posiada sondę emulującą mikrokontroler 80C32 i pracuje pod kontrolą programu przeznaczonego dla systemu MS-DOS. Drugi emulator, pracujący pod kontrolą oprogramowania działającego w systemie Windows, posiada sondę dla mikrokontrolera Dallas 80C320. Sondy obu emulatorów są podłączone do układów testowych. Płytki testowe są wyposażone w dwa układy scalone komunikujące się z mikrokontrolerem za pośrednictwem magistrali I2C oraz w wejścia cyfrowe (przełączniki) i wyjścia sygnalizacyjne (dwukolorowa dioda LED). Układy I2C będą wykorzystane do ćwiczenia 15 - Szeregowe Magistrale Synchroniczne. W poniższej tabeli podano przyporządkowanie wejść i wyjść wyprowadzeniom mikrokomputera jednoukładowego.

Wejście/Wyjście	Element	Port mikrokomputera
Wejście	Przełącznik	P3.5
Wyjście	LED zielona	P1.2
Wyjście	LED czerwona	P1.3

Niniejsze ćwiczenie ma za zadanie zaznajomienie studentów z działaniem emulatora i jego wykorzystania w uruchamianiu prostych programów oraz nabycie podstaw umiejętności pisania programów w języku assemblera procesorów zgodnych z 8051.

Przebieg ćwiczenia.

- Zaznajomienie z podstawowymi rozkazami procesorów 8051.
- Demonstracja prostego programu napisanego w języku assemblera.
- Demonstracja przebiegu edycji i asemlacji programu, omówienie plików wynikowych.
- Demonstracja działania emulatora wraz z oprogramowaniem w systemie DOS i Windows.
- Demonstracja podstawowych funkcji emulatora:
 - o Ładowanie programu
 - o Uruchomienie programu
 - o Praca krokowa
 - o Ustawianie punktów zatrzymania
 - o Przeglądanie zawartości rejestrów
 - o Przeglądanie zawartości pamięci
- Wykonanie zadania podanego przez prowadzącego.
- Kontrola wykonania zadania.

Sprawozdanie.

Sprawozdanie musi zawierać następujące elementy:

- Stronę tytułową zgodną z podanym wzorcem, zawierającą nazwiska wszystkich biorących udział w ćwiczeniu osób.
- Treść zadania podanego przez prowadzącego.
- Program skomentowany będący rozwiązaniem zadania.
- Wnioski z ćwiczenia.

Zaliczenie i ocena ćwiczenia.

Ocena z ćwiczenia jest średnią z dwóch ocen:

- Oceny za sprawozdanie (z wagą 1).
- Oceny ze sprawdzianu pisemnego (z wagą 2).

Sprawdzian obejmuje umiejętność napisania prostego programu assemblerowego dla procesorów zgodnych z 8051.

Prowadzący może oceniać również przygotowanie do ćwiczenia oraz przebieg ćwiczenia.

Literatura.

- Lista rozkazów mikrokomputera jednokładowego serii 8051 - dostępna na laboratorium i na stronie www ZMiTAC.
- Dowolna książka opisująca mikrokomputery serii 8051, przykładowo:
 - o Henryk Małysiak: "Mikrokomputery jednokładowe serii MCS48, MCS51, MCS96", Wydawnictwo Pracowni Jacka Skalmierskiego, Gliwice 1992.
 - o Andrzej Rydzewski: "Mikrokomputery jednokładowe rodziny MCS-51", WNT, Warszawa 1999.