

WOJSKOWA AKADEMIA TECHNICZNA



SYSTEMY WBUDOWANE

Prowadzący: **mgr inż. Waldemar Szyłberg**

Autorzy referatu: **Grzegorz Pol
Kamil Sołowiej
Marcin Werynowski
Maciej Staszczyk
Dawid Wiśniewski**

Grupa szkoleniowa: **I7X3S1**

Numer ćwiczenia: **4.2**

Data laboratorium: **20.05.2009r.**

Data oddania: **01.06.2009r.**

Temat: Badanie mikroprocesora za pomocą analizatora stanów.

Badanie mikroprocesora za pomocą analizatora stanów

1. Specyfikacja szczegółowych zadań określonych przez wykładowców

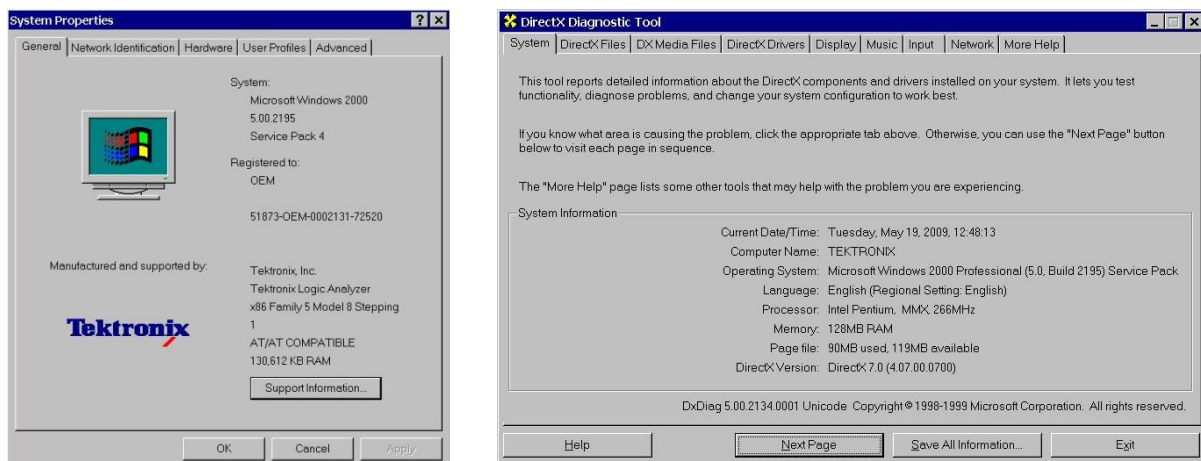
Zgodnie z poleceniem prowadzącego nasze laboratoria mieliśmy do wykonania ćwiczenie oznaczone nr 3 z instrukcji *Quick Start Board*, które to składało się z pięciu powiązanych ze sobą podpunktów, które należało kolejno wykonać. Niestety ze względu na niedostateczną dla nas ilość czasu zdołaliśmy wykonać tylko 4 z nich.

2. Specyfikacja elementów stanowiska laboratoryjnego oraz opis konfiguracji zarówno sprzętu jak i oprogramowania

Do zrealizowania ćwiczenia 4.2 posłużyliśmy się następującymi urządzeniami:

- Analizator stanów logicznych Tektronix TLA-720 wyposażony w dwie karty rozszerzeń TLA 7N3 po 102 kanały pomiarowe
- Sondy P6434(3 szt.)
- Płyta treningowa TLA 7QS
- Monitor Samsung SyncMaster 950p Plus
- Mysz Logitech (3przyciskowa, interfejs PS/2)
- Standardowa klawiatura PC (interfejs PS/2)
- Drukarka Hewlett Packard 1120C (nieużywana podczas wykonania naszego ćwiczenia)

Postanowiliśmy bliżej przyjrzeć się naszemu analizatorowi stanów logicznych. Okazał się on zwykłym komputerem PC (co prawda zamknięty w znacznie innej obudowie) z zainstalowanym systemem operacyjnym Microsoft Windows 2000 z dodatkiem Service Pack 4 (wersja OEM-owa). Z okienka *Właściwości Systemu* odczytaliśmy także ilość pamięci RAM (130,612 KB).



Niestety z wyżej wspomnianego okienka (screen znajduje się powyżej) nie byliśmy w stanie wywnioskować zbyt wiele o procesorze, który jest schowany w obudowie. Żeby zapobiec ponownemu uruchomieniu komputera skorzystaliśmy z *Narzędzia diagnostycznego DirectX*. Tutaj dowiedzieliśmy się, że do dyspozycji mamy 226 Mhz procesor Intel Pentium MMX.

Z przebiegu laboratorium oraz z instrukcji urządzenia dowiedzieliśmy się, że analizator jest łatwym w użyciu instrumentem pomiarowym zawierającym w sobie technologie wielokanałowych wyjść, głębokiej pamięci oraz analizowanie wysokich częstotliwości badanego sygnału. Umożliwia jednoczesną analizę stanów i czasów przełączania w jednej sondzie eliminując potrzebę używania wielu urządzeń sondujących jednocześnie. Dostępny na naszych laboratoriach model TLA 720 wykorzystuje technologię „MagniVu” umożliwiając próbkowanie z rozdzielczością 500 ps na wszystkich kanałach. Wszystkie analizowane dane są próbkowane z

Badanie mikroprocesora za pomocą analizatora stanów

częstotliwością do 2 GHz, w zależności od sposobu użycia urządzenia. Zebrane dane są następnie obrabiane w czasie rzeczywistym uwzględniając najmniejsze zmiany w przebiegach na dowolnym kanale.

Oprogramowanie, które wykorzystaliśmy to *TLA Application* używane do obsługi i wizualizacji analizy. Systemem operacyjnym jak już wyżej wspominałem był Microsoft Windows 2000 SP4.

Płyta treningowa TEKTRONIX TLA 7QS została podłączona do kart rozszerzeń analizatora stanów, a oprogramowanie firmy Tektronix zostało skonfigurowane do łączenia się z płytą treningową TEKTRONIX TLA 7QS.

3. Opis przebiegu ćwiczenia

Celem ćwiczenia było zapoznanie się z możliwościami analizatora sygnałów TLA 720 i typowymi błędami w systemach mikrokomputerowych.

Ćw. 1. Wyzwolenie podczas włączania i przechwytywanie kodu sekwencji startowej.

Cel ćwiczenia: Przechwycenie sekwencji startowych płyty treningowej TEKTRONIX TLA 7QS oraz zapoznanie się z metodą wyzwalania analizatora TLA 720.

Przebieg ćwiczenia:

- » ćwiczenie rozpoczyna się załadowaniem systemu pomiarowego ćwiczenia nr 1,
- » wyłączeniem płyty treningowej,
- » uruchomienie przyciskiem RUN załadowanego oprogramowania,
- » po zmianie RUN na STOP włączyć płytę treningową,
- » analiza uzyskanych wyników.

Ćw. 2. Użycie zegarów wyzwalających do pomiaru opóźnienia wykonania przerwania.

Cel ćwiczenia: Pomiar i rejestracja czasu opóźnienia przy pomocy triggerów między zgłoszeniem przerwania a jego obsłużeniem.

Przebieg ćwiczenia:

- » załadowanie systemu pomiarowego ćwiczenia nr 2,
- » uruchomienie systemu przyciskiem RUN,
- » odczytanie czasu zgłoszenia przerwania (timer 1) i czasu jego obsługi (timer 2),
- » analiza uzyskanych wyników.

Ćw. 3. Wyzwolenie przy zapisywaniu błędnych danych do wyświetlacza LED.

Cel ćwiczenia: Lokalizacja wystąpienia błędu zapisu danych do portu wyświetlacza LED.

Przebieg ćwiczenia:

- » ćwiczenie rozpoczyna się załadowaniem systemu pomiarowego ćwiczenia nr 3,
- » uruchomienie systemu przyciskiem RUN,
- » włączenie na płycie treningowej programu AUTO DELAY,
- » analiza uzyskanych wyników.

Ćw. 4. Wyzwolenie przy niewłaściwym odczytaniu danych przez mikroprocesor.

Cel ćwiczenia: Lokalizacja błędu odczytu danych z pamięci. Wykorzystanie technologii MagniVu.

Przebieg ćwiczenia:

- » ćwiczenie rozpoczyna się załadowaniem systemu pomiarowego ćwiczenia nr 4,
- » wyłączenie programu AUTO DELAY na płycie treningowej,
- » uruchomienie systemu przyciskiem RUN,
- » włączenie na płycie treningowej programu AUTO DELAY,
- » analiza uzyskanych wyników.

4. Opis przebiegu ćwiczenia

Ćw. 1. Wyzwolenie podczas włączania i przechwytywanie kodu sekwencji startowej.

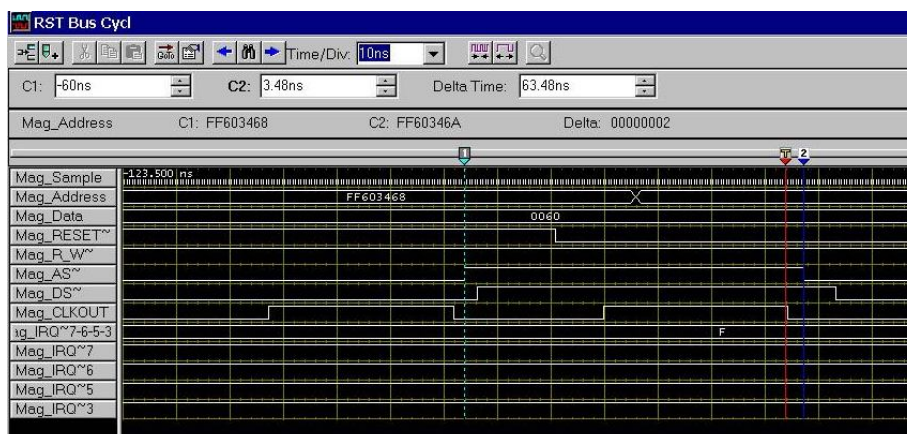
Pierwszy etap polegał na weryfikacji kodu startowego systemu z mikroprocesora. Trigger został ustawiony na przycisk *Power-on Reset*. Na tym etapie przechwytywaliśmy kod generowany przez procesor podczas uruchamiania. W tym celu załadowaliśmy setup z pliku *1-Capture Power Up Code.tla*. Następnie wcisnęliśmy przycisk Run i po zatrzymaniu się programu przeanalizowaliśmy uzyskane wyniki, a także obejrzelśmy triggerry i ustawienia kanałów.

| | | | | | | |
|----|----------|------|----------------------------|-----|------|----------------|
| 1 | 00FFFFFF | FFFF | (RESET) | (S) | 1011 | -33.250,500 us |
| 2 | 00000000 | 0000 | (RESET: STACK POINTER) | (S) | 1011 | 0 ps |
| 3 | 00000002 | 6320 | (RESET: STACK POINTER) | (S) | 1011 | 375,000 ns |
| 4 | 00000004 | 0060 | (RESET: PROGRAM COUNTER) | (S) | 1011 | 750,000 ns |
| 5 | 00000006 | 320A | (RESET: PROGRAM COUNTER) | (S) | 1011 | 1.125,000 us |
| 6 | 0060320A | 7007 | MOVEQ #00000007,D0 | (S) | 1011 | 1.562,500 us |
| 7 | 0060320C | 4E7B | MOVEC D0,DFC | (S) | 1011 | 1.937,500 us |
| 8 | 0060320E | 0001 | (EXTENSION) | (S) | 1011 | 2.312,500 us |
| 9 | 00603210 | 203C | MOVE.L #FFFFFF001,D0 | (S) | 1011 | 2.687,500 us |
| 10 | 00603212 | FFFF | (EXTENSION) | (S) | 1011 | 3.062,500 us |
| 11 | 00603214 | F001 | (EXTENSION) | (S) | 1011 | 3.625,000 us |
| 12 | 00603216 | 0EB9 | MOVES.L D0,0003FF00 | (S) | 1011 | 4.000,000 us |
| 13 | 00603218 | 0800 | (EXTENSION) | (S) | 1011 | 4.375,000 us |
| 14 | 0060321A | 0003 | (EXTENSION) | (S) | 1011 | 4.750,000 us |

Adres na wierzchołku stosu wskazuje adres obsługi urządzenia wbudowanego czyli 0060320A. Flush wystąpił z powodu skoku branch (dlatego trzeba było wyczyścić bufor rozkazów w procesorze potokowym).

| | | | | | | |
|----|----------|------|----------------------|-----|------|---------------|
| 50 | 00609332 | FFFD | (READ) | (S) | 1011 | 21.438,000 us |
| 51 | 00609334 | 0020 | (READ) | (S) | 1011 | 22.375,500 us |
| 52 | 00609336 | 0001 | (READ) | (S) | 1011 | 22.750,500 us |
| 53 | 00609338 | 001F | (READ) | (S) | 1011 | 23.688,000 us |
| 54 | 0060933A | FFFF | (READ) | (S) | 1011 | 24.063,000 us |
| 55 | 0060933C | 0040 | (READ) | (S) | 1011 | 25.000,500 us |
| 56 | 0060933E | 0001 | (READ) | (S) | 1011 | 25.375,500 us |
| 57 | 0060324A | 0000 | (EXTENSION) | (S) | 1011 | 26.188,000 us |
| 58 | 0060324C | 6320 | (EXTENSION) | (S) | 1011 | 26.563,000 us |
| 59 | 0060324E | 2C7C | MOVEA.L #00000000,A6 | (S) | 1011 | 26.938,000 us |
| 60 | 00603250 | 0000 | (EXTENSION) | (S) | 1011 | 27.313,000 us |
| 61 | 00603252 | 0000 | (EXTENSION) | (S) | 1011 | 27.688,000 us |
| 62 | 00603254 | 227C | MOVEA.L #00001000,A1 | (S) | 1011 | 28.063,000 us |
| 63 | 00603256 | 0000 | (EXTENSION) | (S) | 1011 | 28.438,000 us |
| 64 | 00603258 | 1000 | (EXTENSION) | (S) | 1011 | 28.813,000 us |
| 65 | 0060325A | 203C | MOVE.L #00000320,D0 | (S) | 1011 | 29.188,000 us |
| 66 | 0060325C | 0000 | (EXTENSION) | (S) | 1011 | 29.563,000 us |
| 67 | 0060325E | 0320 | (EXTENSION) | (S) | 1011 | 29.938,000 us |
| 68 | 00603260 | E488 | LSR.L #2,D0 | (S) | 1011 | 30.313,000 us |
| 69 | 00603262 | 6002 | BRA.B 00603266 | (S) | 1011 | 30.688,000 us |
| 70 | 00603264 | 4299 | (FLUSH) | (S) | 1011 | 31.063,000 us |
| 71 | 00603266 | 51C8 | DBF D0,00603264 | (S) | 1011 | 31.438,000 us |
| 72 | 00603268 | FFFF | (EXTENSION) | (S) | 1011 | 31.813,000 us |
| 73 | 0060326A | 4299 | CLR.L (A1)+ | (S) | 1011 | 32.500,500 us |
| 74 | 00603266 | 51C8 | DBF D0,00603264 | (S) | 1011 | 32.875,500 us |
| 75 | 00603268 | FFFF | (EXTENSION) | (S) | 1011 | 33.250,500 us |

Mogliśmy też obejrzeć przebiegi czasowe dowolnych linii za pomocą okna *RST Bus Cycl Waveform*:

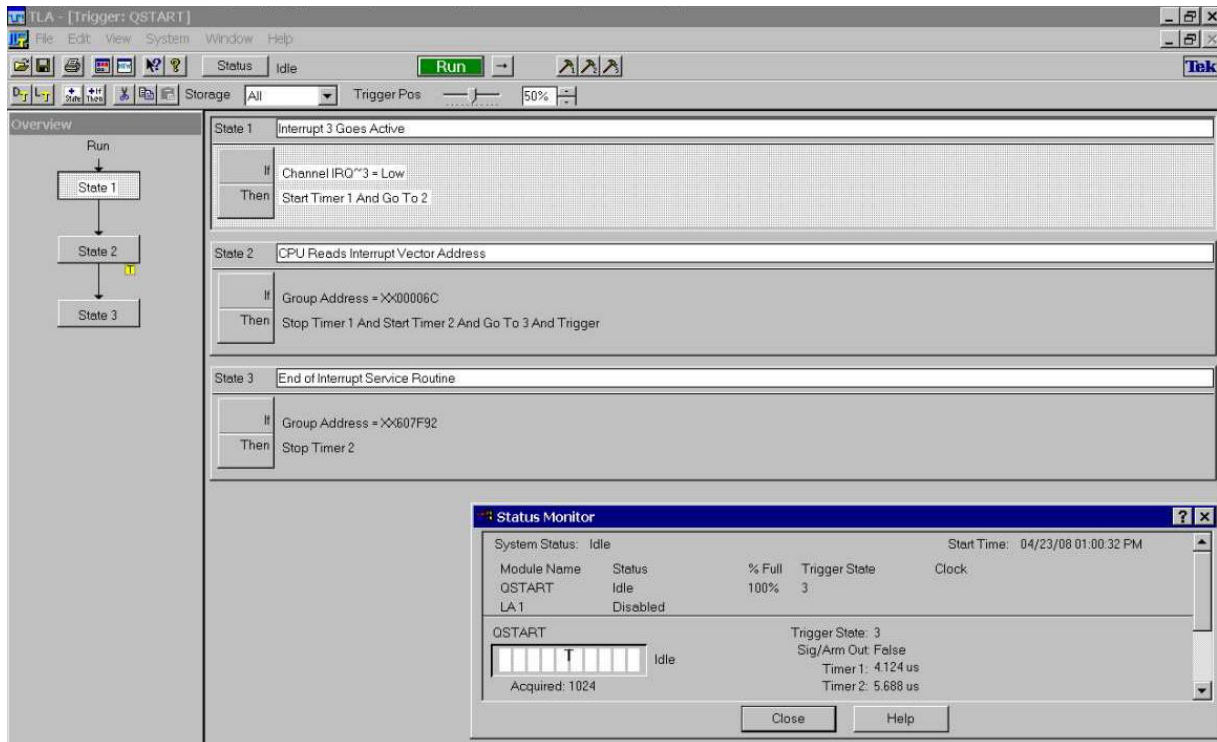


Badanie mikroprocesora za pomocą analizatora stanów

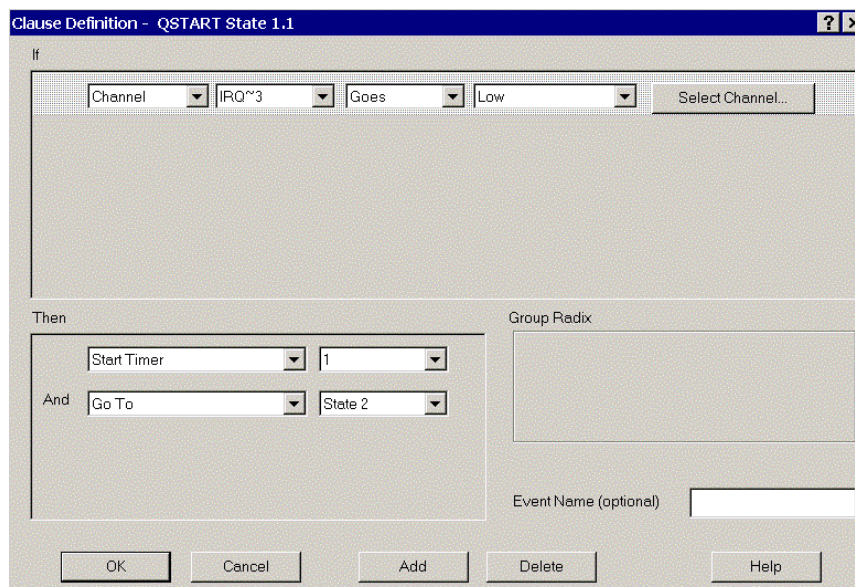
Na podstawie tych wykresów i za pomocą suwaków mierzących czas między fazami na wykresie, mogliśmy odczytać okres taktowania procesora. Odczytaliśmy dość dokładny wynik 63,48ns. Z tego wyniku możemy obliczyć częstotliwość taktowania procesora, która wynosi 15.75MHz.

Ćw. 2. Użycie zegarów wyzwalających do pomiaru opóźnienia wykonania przerwania.

Czas wykonywania procedury obsługi przerwania możemy zmierzyć ustawiając wyzwalacz(Trigger), tak aby zatrzymał wcześniej wystartowany timer. Okno Trigger: QSTART służy do ustawiania triggerów. Przejrzyście obrazuje w którym momencie nastąpi wyzwolenie, oraz przy jakich okolicznościach. Poniżej widać okno *Status Monitor*, pokazujący stan wyzwalaczy, oraz timery.



Okno *Clause Definition* pozwala na określenie i zmianę okoliczności wyzwolenia. Z rozwijanych list można ustawić kiedy będzie następowało wyzwolenie i z jakiego powodu.



Badanie mikroprocesora za pomocą analizatora stanów

Poniżej zamieściłem okno kodu oraz *Status Monitor*, pokazujący czas opóźnienia przerwania. Jest to czas zmierzony prawidłowo ustawionym triggerem, który to pokazał nam, że ten czas wynosi 5.688us

The screenshot shows the TLA - [Int Svc Code] debugger interface. The main window displays assembly code with columns for Sample, Q-Start Address, Q-Start Data, Q-Start Mnemonic, IRQ-7-6-5-3, and Timestamp. A red arrow points to sample 511, which is the start of the interrupt. The Status Monitor window is overlaid on the code, showing system status (Idle), module name (QSTART), and a timer showing 5.688 us. The Status Monitor also displays a list of modules and their status, and a trigger state indicator.

| Sample | Q-Start Address | Q-Start Data | Q-Start Mnemonic | IRQ-7-6-5-3 | Timestamp |
|--------|-----------------|--------------|------------------------|-------------|---------------|
| 491 | 0060346A | 3540 | (EXTENSION) | (S) 1111 | -8.125,000 us |
| 492 | 0060346C | 4EF9 | (FLUSH) | (S) 1111 | -7.750,000 us |
| 493 | 00603540 | 41F8 | LEA FFFFFFF00, A0 | (S) 1111 | -7.375,000 us |
| 494 | 00603542 | F700 | (EXTENSION) | (S) 1111 | -7.000,000 us |
| 495 | 00603544 | 7000 | MOVEQ #00000000, D0 | (S) 1111 | -6.625,000 us |
| 496 | 00603546 | 0828 | BTST.B #00, (0019, A0) | (S) 1111 | -6.250,000 us |
| 497 | 00603548 | 0000 | (EXTENSION) | (S) 1111 | -5.875,000 us |
| 498 | 0060354A | 0019 | (EXTENSION) | (S) 1111 | -5.500,000 us |
| 499 | 0060354C | 8702 | BEQ.B #0000550 | (S) 1111 | -5.125,000 us |
| 500 | 0060354E | 7001 | MOVEQ #00000001, D0 | (S) 1111 | -4.750,000 us |
| 501 | 00603550 | 4E75 | RTS | (S) 1111 | -4.375,000 us |
| 502 | 00603552 | 41F8 | (FLUSH) | (S) 1111 | -4.000,000 us |
| 503 | 000062C0 | 0060 | (READ) | (S) 1110 | -3.625,000 us |
| 504 | 000062C2 | 7E7C | (WRITE) | (S) 1110 | -3.250,000 us |
| 505 | 00607E7C | 4A80 | TST.L D0 | (S) 1110 | -2.875,000 us |
| 506 | 00FFFFFF7 | 4A80 | (INT ACK LEVEL: 3) | (S) 1110 | -2.500,000 us |
| 507 | 000062C2 | 006C | (WRITE) | (S) 1110 | -2.125,000 us |
| 508 | 000062BC | 2004 | (WRITE) | (S) 1110 | -1.750,000 us |
| 509 | 000062BE | 0060 | (WRITE) | (S) 1110 | -1.375,000 us |
| 510 | 000062C0 | 7E7C | (WRITE) | (S) 1110 | -1.000,000 us |
| 511 | 0000006C | 0060 | (IPL 3 AUTOVECTOR) | (S) 1110 | 0 ps |
| 512 | 0000006E | 7F80 | (IPL 3 AUTOVECTOR) | (S) 1110 | 375,500 ns |
| 513 | 00607F80 | 2F00 | MOVE.L D0, -(A7) | (S) 1110 | 812,500 ns |
| 514 | 00607F82 | 7005 | MOVEQ #00000005, D0 | (S) 1110 | 1,187,500 us |
| 515 | 000062B8 | 0000 | (WRITE) | (S) 1110 | 1,562,500 us |
| 516 | 000062BA | 0000 | (WRITE) | (S) 1110 | 1,937,500 us |
| 517 | 00607F84 | 23C0 | MOVE.L D0, 0000130C | (S) 1110 | 2,312,500 us |
| 518 | 00607F86 | 0000 | (EXTENSION) | (S) 1110 | 2,687,500 us |
| 519 | 00607F88 | 130C | (EXTENSION) | (S) 1110 | 3,062,500 us |
| 520 | 00607F8A | 42B9 | CLR.L 00001310 | (S) 1110 | 3,437,500 us |
| 521 | 00607F8C | 0000 | (EXTENSION) | (S) 1110 | 3,812,500 us |
| 522 | 0000130C | 0000 | (WRITE) | (S) 1110 | 4,187,500 us |
| 523 | 0000130E | 0005 | (WRITE) | (S) 1110 | 4,562,500 us |
| 524 | 00607F8E | 1310 | (EXTENSION) | (S) 1110 | 4,937,500 us |
| 525 | 00607F90 | 201E | MOVE.L (A7)+, D0 | (S) 1110 | 5,312,500 us |
| 526 | 00607F92 | 4E73 | RTS | (S) 1110 | 5,687,500 us |
| 527 | 00001310 | 0000 | (WRITE) | (S) 1110 | 6,062,500 us |
| 528 | 00001312 | 0000 | (WRITE) | (S) 1110 | 6,437,500 us |
| 529 | 000062B8 | 0000 | (READ) | (S) 1110 | 6,812,500 us |
| 530 | 000062BA | 0000 | (READ) | (S) 1110 | 7,187,500 us |
| 531 | 00607F94 | 4E73 | (FLUSH) | (S) 1110 | 7,562,500 us |
| 532 | 000062BC | 2004 | (READ) | (S) 1110 | 7,937,500 us |
| 533 | 000062C2 | 006C | (READ) | (S) 1110 | 8,312,500 us |
| 534 | 000062BE | 0060 | (READ) | (S) 1110 | 8,687,500 us |
| 535 | 000062C0 | 7E7C | (READ) | (S) 1110 | 9,062,500 us |
| 536 | 00607E7C | 4A80 | TST.L D0 | (S) 1110 | 9,437,500 us |
| 537 | 00FFFFFF7 | 4A80 | (INT ACK LEVEL: 3) | (S) 1110 | 9,812,500 us |
| 538 | 000062C2 | 006C | (WRITE) | (S) 1110 | 10,187,500 us |
| 539 | 000062BC | 2004 | (WRITE) | (S) 1110 | 10,562,500 us |
| 540 | 000062BE | 0060 | (WRITE) | (S) 1110 | 10,937,500 us |
| 541 | 000062C0 | 7E7C | (WRITE) | (S) 1110 | 11,312,500 us |
| 542 | 0000006C | 0060 | (IPL 3 AUTOVECTOR) | (S) 1110 | 11,687,500 us |
| 543 | 0000006E | 7F80 | (IPL 3 AUTOVECTOR) | (S) 1110 | 12,062,500 us |

Ćw. 3. Wyzwolenie przy zapisywaniu błędnych danych do wyświetlacza LED.

W następnym ćwiczeniu tworzyliśmy wyzwalacz uruchamiany przy błędzie zapisu do ekranu LED. Miało to na celu zademonstrowanie używania tego zestawu narzędzi do wykrywania problemów programowych lub sprzętowych. W ćwiczeniu był wykorzystywany program *AUTO DELAY* wgrany na płytę TLA7QSc, który symuluje sprzętowy problem odczytania danych, a ten z kolei powoduje nieprawidłowy wynik na wyświetlaczu LED. Wyzwalacz został zdefiniowany w następujący sposób:

The screenshot shows the System debugger interface. The Trigger configuration window is open, showing the trigger state (Run) and the trigger definition for State 1. The trigger is defined as a Register Read Error. The Clause Definition window is also open, showing the trigger definition: If Group Address = XX440000 And Group Data Is Not AAAA. The System window shows the Logic Analyzer and the LED Error output.

Badanie mikroprocesora za pomocą analizatora stanów

Jak widać zdefiniowano regułę, że jeśli pod adresem xx440000 jest zapisywana wartość inna, niż AAAAh, to jest uruchamiany wyzwalacz. Po uruchomieniu programu z tak ustawioną konfiguracją analizatora, uzyskaliśmy wyniki. Na poniższym wycinku ekranu widać sekwencję instrukcji, która zapisuje dane pod adres 0044 0000 (port wyświetlacza LED). Daną tą powinno być AAAA, jednak, jak widać poniżej, jest nią 8AAA.

| | | | | | | |
|-----|----------|------|---------------------|-----|------|---------------|
| 499 | 00607D36 | 2E1F | { FLUSH } | (S) | 1111 | -4.312,500 us |
| 500 | 0000620E | 0060 | { WRITE } | (S) | 1111 | -3.937,500 us |
| 501 | 00006210 | 7D36 | { WRITE } | (S) | 1111 | -3.562,500 us |
| 502 | 006041A6 | 302F | MOVE.W (0006,A7),D0 | (S) | 1111 | -3.187,500 us |
| 503 | 006041A8 | 0006 | { EXTENSION } | (S) | 1111 | -2.812,500 us |
| 504 | 006041AA | 4640 | NOT.W D0 | (S) | 1111 | -2.437,500 us |
| 505 | 00006214 | 7555 | { READ } | (S) | 1111 | -2.062,500 us |
| 506 | 006041AC | 33C0 | MOVE.W D0,00440000 | (S) | 1111 | -1.687,500 us |
| 507 | 006041AE | 0044 | { EXTENSION } | (S) | 1111 | -1.312,500 us |
| 508 | 006041B0 | 0000 | { EXTENSION } | (S) | 1111 | -937,500 ns |
| 509 | 006041B2 | 205F | MOVEA.L (A7)+,A0 | (S) | 1111 | -562,500 ns |
| 510 | 006041B4 | 588F | ADDQ.L #4,A7 | (S) | 1111 | -187,500 ns |
| 511 | 00440000 | 8AAA | { WRITE } | (S) | 1111 | 0 ps |
| 512 | 0000620E | 0060 | { READ } | (S) | 1111 | 375,000 ns |
| 513 | 00006210 | 7D36 | { READ } | (S) | 1111 | 750,000 ns |
| 514 | 006041B6 | 4ED0 | JMP (A0) | (S) | 1111 | 1.125,000 us |
| 515 | 006041B8 | 204F | { FLUSH } | (S) | 1111 | 1.500,000 us |
| 516 | 00607D36 | 2E1F | MOVE.L (A7)+,D7 | (S) | 1111 | 1.875,000 us |
| 517 | 00607D38 | 4E75 | RTS | (S) | 1111 | 2.250,000 us |
| 518 | 00006216 | 0000 | { READ } | (S) | 1111 | 2.625,000 us |
| 519 | 00006218 | 0084 | { READ } | (S) | 1111 | 3.000,000 us |
| 520 | 00607D3A | 48E7 | { FLUSH } | (S) | 1111 | 3.375,000 us |

Ćw. 4. Wyzwolenie przy niewłaściwym odczycianiu danych przez mikroprocesor.

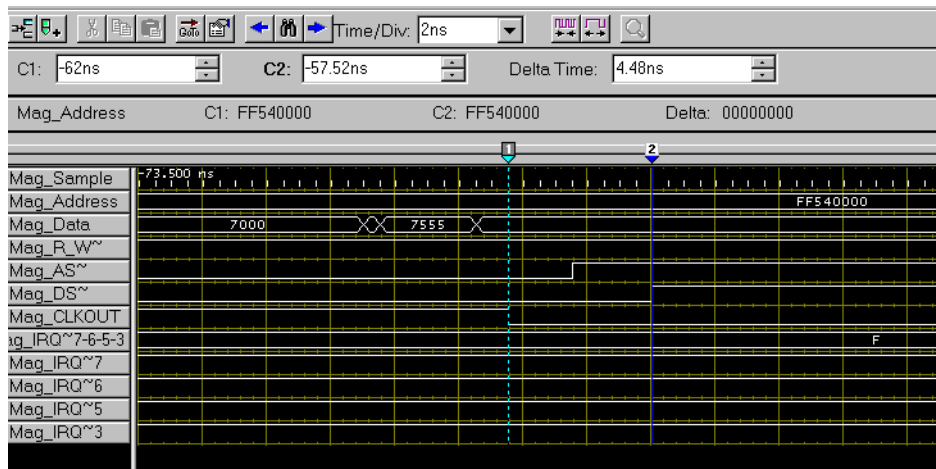
W ćwiczeniu nr 3 użyliśmy analizatora stanów logicznych, aby zauważyć, że dane wpisywane do rejestru wyświetlacza LED były niepoprawne. Mamy jednak pewność, że problem ten nie jest spowodowany błędem sprzętowym, ponieważ wyświetlacz LED okazjonalnie wyświetla poprawne dane.

W tym podpunkcie użyliśmy analizatora logiki oraz zdefiniowanego w nim wyzwalacza, aby przechwycić błąd odczytu z rejestru. Podobnie jak w podpunkcie nr 3 pobieramy dane za pomocą sondy z płyty podczas uruchomienia na niej programu AUTO DELAY. Na poniższym rzucie ekranu widzimy okno zawierające listing zdezasemblowanych operacji wykonywanych przez procesor w otoczeniu błędu odczytu. Dane odczytane spod adresu 054 0000 nie miały wartości 5555 jak oczekiwano, co tłumaczy czemu dane przesłane do rejestru wyświetlacza były błędne.

| | | | | | | |
|-----|----------|------|-----------------------|-----|------|---------------|
| 489 | 00607D14 | 205F | MOVEA.L (A7)+,A0 | (S) | 1111 | -7.937,500 us |
| 490 | 00607D16 | 508F | ADDQ.L #8,A7 | (S) | 1111 | -7.562,500 us |
| 491 | 00006212 | 0060 | { READ } | (S) | 1111 | -7.187,500 us |
| 492 | 00006214 | 7656 | { READ } | (S) | 1111 | -6.812,500 us |
| 493 | 00607D18 | 4ED0 | JMP (A0) | (S) | 1111 | -6.437,500 us |
| 494 | 00607D1A | 2F07 | { FLUSH } | (S) | 1111 | -6.062,500 us |
| 495 | 00607656 | 6100 | BSR.W 00607D1A | (S) | 1111 | -5.687,000 us |
| 496 | 00607658 | 06C2 | { EXTENSION } | (S) | 1111 | -5.312,500 us |
| 497 | 0000621A | 0060 | { WRITE } | (S) | 1111 | -4.875,000 us |
| 498 | 0000621C | 765A | { WRITE } | (S) | 1111 | -4.500,000 us |
| 499 | 00607D1A | 2F07 | MOVE.L D7,-(A7) | (S) | 1111 | -4.125,000 us |
| 500 | 00607D1C | 33FC | MOVE.W #5555,00540000 | (S) | 1111 | -3.750,000 us |
| 501 | 00006216 | 0000 | { WRITE } | (S) | 1111 | -3.375,000 us |
| 502 | 00006218 | 0080 | { WRITE } | (S) | 1111 | -3.000,000 us |
| 503 | 00607D1E | 5555 | { EXTENSION } | (S) | 1111 | -2.625,000 us |
| 504 | 00607D20 | 0054 | { EXTENSION } | (S) | 1111 | -2.250,000 us |
| 505 | 00607D22 | 0000 | { EXTENSION } | (S) | 1111 | -1.875,000 us |
| 506 | 00607D24 | 3E39 | MOVE.W 00540000,D7 | (S) | 1111 | -1.500,000 us |
| 507 | 00540000 | 5555 | { WRITE } | (S) | 1111 | -1.312,500 us |
| 508 | 00607D26 | 0054 | { EXTENSION } | (S) | 1111 | -937,000 ns |
| 509 | 00607D28 | 0000 | { EXTENSION } | (S) | 1111 | -562,500 ns |
| 510 | 00607D2A | 7000 | MOVEQ #00000000,D0 | (S) | 1111 | -187,500 ns |
| 511 | 00540000 | 7555 | { READ } | (S) | 1111 | 0 ps |
| 512 | 00607D2C | 3007 | MOVE.W D7,D0 | (S) | 1111 | 375,000 ns |
| 513 | 00607D2E | 2F00 | MOVE.L D0,-(A7) | (S) | 1111 | 750,000 ns |
| 514 | 00607D30 | 4EB9 | JSR 006041A6 | (S) | 1111 | 1.125,000 us |
| 515 | 00006212 | 0000 | { WRITE } | (S) | 1111 | 1.500,000 us |
| 516 | 00006214 | 7050 | { WRITE } | (S) | 1111 | 1.875,000 us |
| 517 | 00607D32 | 0060 | { EXTENSION } | (S) | 1111 | 2.250,000 us |
| 518 | 00607D34 | 41A6 | { EXTENSION } | (S) | 1111 | 2.625,500 us |
| 519 | 00607D36 | 2E1F | { FLUSH } | (S) | 1111 | 3.000,000 us |
| 520 | 0000620E | 0060 | { WRITE } | (S) | 1111 | 3.375,500 us |
| 521 | 00006210 | 7D36 | { WRITE } | (S) | 1111 | 3.750,000 us |
| 522 | 006041A6 | 302F | MOVE.W (0006,A7),D0 | (S) | 1111 | 4.125,000 us |
| 523 | 006041A8 | 0006 | { EXTENSION } | (S) | 1111 | 4.500,000 us |
| 524 | 006041AA | 4640 | NOT.W D0 | (S) | 1111 | 4.875,000 us |

Badanie mikroprocesora za pomocą analizatora stanów

Zgodnie ze specyfikacją mikroprocesora, dane muszą być wystawione na linii przez minimum 5 ns zanim rozpocznie się opadające zbocze sygnału zegarowego zanim sygnał strobulujący danych zacznie narastać. Na poniższym zrzucie ekranu można zauważyć, że zmierzony czas odzwierciedlający stabilne dane wynosi 4.48 ns, a więc wartość długości badanego odcinka czasu nie jest wystarczająca aby spełnić wymagania specyfikacyjne.



5. Wnioski

Wykonane laboratorium pozwoliło nam się zapoznać z ważnym narzędziem używanym podczas badania i diagnozowania sprzętu komputerowego – analizatorem logicznym. Jak już wspominałem całe ćwiczenie, mające na celu pokazanie najważniejszych cech tego typu urządzeń, było podzielone na pięć kroków, które należało kolejno wykonywać. Podczas trwania laboratorium udało nam się przeprowadzić cztery z tych pięciu kroków.

Wykonując kolejne podpunkty zadania udało nam się w praktyce, a nie tylko na podstawie dokumentacji, poznać elastyczność niektórych z takich urządzeń. Poprzez możliwość dostosowania znaczenia poszczególnych kanałów można stworzyć reguły badania dla różnorodnego sprzętu. Jednocześnie, dzięki możliwości zapisywania tego typu konfiguracji, można stosować predefiniowane ustawienia dla najpopularniejszych konfiguracji sprzętowych.

Od strony możliwości funkcjonalnych, stosowany przez nas program Logic Analyzer dawał duże możliwości na konfigurowanie zależności pomiędzy różnymi wyzwalaczami oraz na definiowanie różnych skutków napotkanych zdarzeń. Można w nim zaprogramować prostą maszynę stanów, pracującą w różny sposób w zależności od historii zdarzeń które wystąpiły, można też wykonać pomiary zależności pomiędzy poszczególnymi zdarzeniami, takie jak odstęp czasu pomiędzy ich wystąpieniami, który jest rejestrowany i prezentowany na koniec przeprowadzonego doświadczenia.

Cała konfiguracja programu może być przeprowadzana poprzez graficzny interfejs Użytkownika, który pozwala na dość wygodną obsługę tak zaawansowanego urządzenia jakim jest analizator stanów TLA-720.