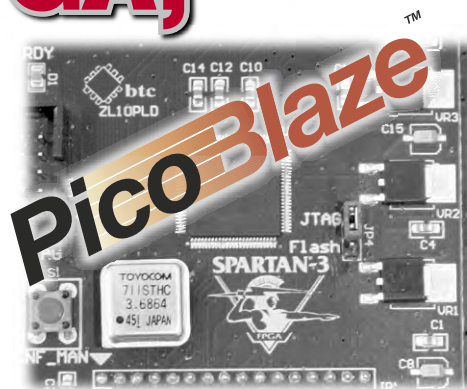


# Zrób sobie procesor: PicoBlaze w FPGA, część 2

## Budowa PicoBlaze

*W drugiej części cyklu omawiamy listę obsługiwanymi instrukcji i przedstawiamy podstawowe informacje o budowie mikroprocesora PicoBlaze.*



Jak już wspomnieliśmy, PicoBlaze został przygotowany przez Kena Chapmana, inżyniera z firmy Xilinx. Pliki źródłowe mikroprocesora są dostępne na stronach internetowych tej firmy. Aby móc je ściągnąć należy uprzednio zarejestrować się. Obecnie są dostępne trzy wersje PicoBlaze, z których w przykładach wykorzystamy najnowszą (przewidzianą do implementacji w układach Spartan-3, Virtex-4, Virtex-II, Virtex-II Pro i nowszych, np. Virtex-5).

### Lista rozkazów PicoBlaze

W tab. 2 umieszczono alfabetyczny spis rozkazów PicoBlaze. Rozkazy podano w formie rozpoznawanej przez kompilator assemblera *kcpsm3.exe*. Nazwy instrukcji podane w nawiasach

są rozpoznawane przez symulator wbudowany w środowisko pBlazIDE.

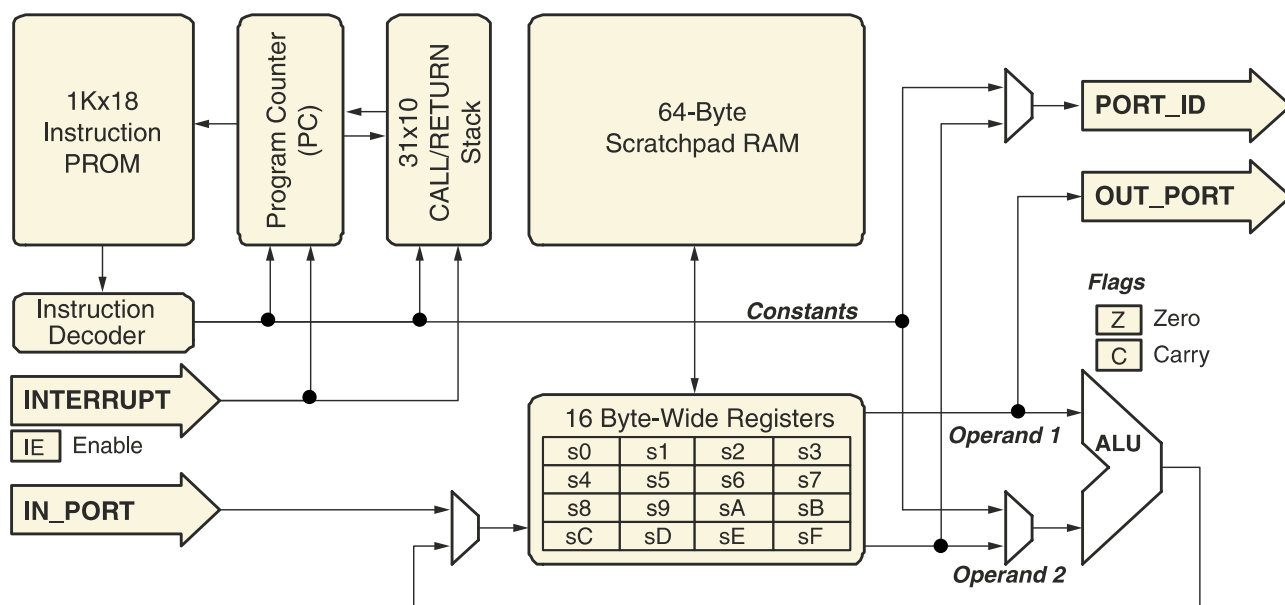
### PicoBlaze w detalach

Schemat blokowy mikroprocesora PicoBlaze pokazano na rys. 4. Wyposażono go w szesnaście 8-bitowych rejestrów ogólnego przeznaczenia. Są one ponumerowane od s0 do sF. Nie ma wyróżnionych rejestrów przeznaczonych do realizacji specjalnych zadań. Również nie ma wyszczególnionego rejestru jako akumulator. Każdy z 16 rejestrów może pełnić taką funkcję.

Zastosowana w PicoBlaze jednostka arytmetyczno-logiczna to standardowa jednostka przeznaczona do wykonywania operacji arytmetyczno-logicznych na danych przechowywanych w rej-

strach ośmiobitowych. Wszystkie operacje są wykonywane na rejestrach (sX), wyspecyfikowanych w instrukcji. Wynik operacji jest przekazywany do tego samego rejestru, jeśli wymaga tego instrukcja. Operacje dwuargumentowe – jako drugi operand (sY) może zostać podany inny rejestr lub stała (kk), również ośmiobitowa. Możliwość podania stałej w argumencie powoduje zwiększenie listy rozkazów. Dla operacji dodawania i odejmowania, które wymagają większej szerokości argumentów przewidziano możliwość wykorzystania bitu przeniesienia. Nie ma operacji mnożenia i dzielenia.

Mikroprocesor wyposażono w dwa znaczniki (flagi): zera (ZERO) i przeniesienia (CARRY). Znaczniki te są wykorzystywane są m.in. do wyko-



Rys. 4. Schemat blokowy mikroprocesora PicoBlaze

Tab. 2.

Instrukcja	Opis	Funkcja	ZERO	CARRY
ADD sX, kk	Suma rejestru sX i stałej kk	$sX \leftarrow sX + kk$	?	?
ADD sX, sY	Suma rejestru sX i rejestru sY	$sX \leftarrow sX + sY$	?	?
ADDCY sX, kk (ADDC)	Suma rejestru sX i stałej kk z bitem przeniesienia CARRY	$sX \leftarrow sX + kk + CARRY$	?	?
ADDCY sX, sY (ADDC)	Suma rejestru sX i rejestru sY z bitem przeniesienia CARRY	$sX \leftarrow sX + sY + CARRY$	?	?
AND sX, kk	Logiczny iloczyn bitów rejestru sX i stałej kk	$sX \leftarrow sX \text{ AND } kk$	?	0
AND sX, sY	Logiczny iloczyn bitów rejestru sX i rejestru sY	$sX \leftarrow sX \text{ AND } sY$	?	0
CALL aaa	Skok do procedury pod aaa	$TOS \leftarrow PC$ $PC \leftarrow aaa$	-	-
CALL C, aaa	Jeśli flaga CARRY jest ustawiona, skok do procedury pod aaa	Jeśli $CARRY = 1$ , { $TOS \leftarrow PC$ , $PC \leftarrow aaa$ }	-	-
CALL NC, aaa	Jeśli flaga CARRY nie jest ustawiona, skok do procedury pod aaa	Jeśli $CARRY = 0$ , { $TOS \leftarrow PC$ , $PC \leftarrow aaa$ }	-	-
CALL NZ, aaa	Jeśli flaga ZERO nie jest ustawiona, skok do procedury pod aaa	Jeśli $ZERO = 0$ , { $TOS \leftarrow PC$ , $PC \leftarrow aaa$ }	-	-
CALL Z, aaa	Jeśli flaga ZERO jest ustawiona, skok do procedury pod aaa	Jeśli $ZERO = 1$ , { $TOS \leftarrow PC$ , $PC \leftarrow aaa$ }	-	-
COMPARE sX, kk (COMP)	Porównaj rejestr sX i stałą kk, ustaw odpowiednio flagi CARRY i ZERO	Jeśli $sX = kk$ , $ZERO \leftarrow 1$ Jeśli $sX < kk$ , $CARRY \leftarrow 1$	?	?
COMPARE sX, sY (COMP)	Porównaj rejestr sX i rejestr sY, ustaw odpowiednio flagi CARRY i ZERO	Jeśli $sX = sY$ , $ZERO \leftarrow 1$ Jeśli $sX < sY$ , $CARRY \leftarrow 1$	?	?
DISABLE INTERRUPT (DINT)	Zablokuj wejście przerwania	$INTERRUPT\_ENABLE \leftarrow 0$	-	-
ENABLE INTERRUPT (EINT)	Odblokuj wejście przerwania	$INTERRUPT\_ENABLE \leftarrow 1$	-	-
FETCH sX, (sY) (FETCH sX, sY)	Odczytaj zawartość pamięci podręcznej RAM, wskazywanej przez rejestr sY, do rejestru sX	$sX \leftarrow RAM[(sY)]$	-	-
FETCH sX, ss	Odczytaj zawartość pamięci podręcznej RAM, wskazywanej przez wartość ss, do rejestru sX	$sX \leftarrow RAM[ss]$	-	-
INPUT sX, (sY) (IN sX, sY)	Odczytaj zawartość portu wejściowego, wskazywanego przez rejestr sY, do rejestru sX	$PORT\_ID \leftarrow sY$ $sX \leftarrow IN\_PORT$	-	-
INPUT sX, pp (IN sX, pp)	Odczytaj zawartość portu wejściowego, wskazywanego przez wartość pp, do rejestru sX	$PORT\_ID \leftarrow pp$ $sX \leftarrow IN\_PORT$	-	-
JUMP aaa	Bezwarunkowy skok programu do adresu aaa	$PC \leftarrow aaa$	-	-
JUMP C, aaa	Jeśli flaga CARRY jest ustawiona, to skok programu do adresu aaa	Jeśli $CARRY = 1$ , $PC \leftarrow aaa$	-	-
JUMP NC, aaa	Jeśli flaga CARRY nie jest ustawiona, to skok programu do adresu aaa	Jeśli $CARRY = 0$ , $PC \leftarrow aaa$	-	-
JUMP NZ, aaa	Jeśli flaga ZERO nie jest ustawiona, to skok programu do adresu aaa	Jeśli $ZERO = 0$ , $PC \leftarrow aaa$	-	-
JUMP Z, aaa	Jeśli flaga ZERO jest ustawiona, to skok programu do adresu aaa	Jeśli $ZERO = 1$ , $PC \leftarrow aaa$	-	-
LOAD sX, kk	Wczytaj rejestr sX wartością kk	$sX \leftarrow kk$	-	-
LOAD sX, sY	Wczytaj rejestr sX wartością rejestru sY	$sX \leftarrow sY$	-	-
OR sX, kk	Logiczna suma bitów rejestru sX i stałej kk	$sX \leftarrow sX \text{ OR } kk$	?	0
OR sX, sY	Logiczna suma bitów rejestru sX i rejestru sY	$sX \leftarrow sX \text{ OR } sY$	?	0
RETURN (RET)	Bezwarunkowy powrót z procedury	$PC \leftarrow TOS + 1$	-	-
RETURN C (RET C)	Jeśli flaga CARRY jest ustawiona, to powrót z procedury	Jeśli $CARRY = 1$ , $PC \leftarrow TOS + 1$	-	-
RETURN NC (RET NC)	Jeśli flaga CARRY nie jest ustawiona, to powrót z procedury	Jeśli $CARRY = 0$ , $PC \leftarrow TOS + 1$	-	-
RETURN NZ (RET NZ)	Jeśli flaga ZERO nie jest ustawiona, to powrót z procedury	Jeśli $ZERO = 0$ , $PC \leftarrow TOS + 1$	-	-
RETURN Z (RET Z)	Jeśli flaga ZERO jest ustawiona, to powrót z procedury	Jeśli $ZERO = 1$ , $PC \leftarrow TOS + 1$	-	-
RETURN DISABLE (RETI DISABLE)	Powrót z procedury obsługi przerwania, przerwanie pozostaje zablokowane	$PC \leftarrow TOS$ $ZERO \leftarrow$ zachowane ZERO $CARRY \leftarrow$ zachowane CARRY $INTERRUPT\_ENABLE \leftarrow 0$	?	?
RETURN ENABLE (RETI ENABLE)	Powrót z procedury obsługi przerwania, aktywowanie przerwania	$PC \leftarrow TOS$ $ZERO \leftarrow$ zachowane ZERO $CARRY \leftarrow$ zachowane CARRY $INTERRUPT\_ENABLE \leftarrow 1$	?	?
RL sX	Obróć bity rejestr sX w lewo o jedną pozycję	$sX \leftarrow \{sX[6:0], sX[7]\}$ $CARRY \leftarrow sX[7]$	?	?

Tab. 2. c.d.

Instrukcja	Opis	Funkcja	ZERO	CARRY
RR sX	Obróć bity rejestru sX w prawo o jedną pozycję	$sX \leftarrow \{sX[0], sX[7:1]\}$ $CARRY \leftarrow sX[0]$	?	?
SLO sX	Obróć bity rejestru sX w lewo o jedną pozycję, uzupełnij wartością '0'	$sX \leftarrow \{sX[6:0], 0\}$ $CARRY \leftarrow sX[7]$	?	?
SL1 sX	Obróć bity rejestru sX w lewo o jedną pozycję, uzupełnij wartością '1'	$sX \leftarrow \{sX[6:0], 1\}$ $CARRY \leftarrow sX[7]$	?	?
SLA sX	Obróć bity rejestru sX w lewo o jedną pozycję, uzupełnij wartością flagi CARRY	$sX \leftarrow \{sX[6:0], CARRY\}$ $CARRY \leftarrow sX[7]$	?	?
SLX sX	Obróć bity rejestru sX w lewo o jedną pozycję, nie zmieniaj wartości bitu sX[0]	$sX \leftarrow \{sX[6:0], sX[0]\}$ $CARRY \leftarrow sX[7]$	?	?
SR0 sX	Obróć bity rejestru sX w prawo o jedną pozycję, uzupełnij wartością '0'	$sX \leftarrow \{0, sX[7:1]\}$ $CARRY \leftarrow sX[0]$	?	?
SR1 sX	Obróć bity rejestru sX w prawo o jedną pozycję, uzupełnij wartością '1'	$sX \leftarrow \{1, sX[7:1]\}$ $CARRY \leftarrow sX[0]$	?	?
SRA sX	Obróć bity rejestru sX w prawo o jedną pozycję, uzupełnij wartością flagi CARRY	$sX \leftarrow \{CARRY, sX[7:1]\}$ $CARRY \leftarrow sX[0]$	?	?
SRX sX	Obróć bity rejestru sX w prawo o jedną pozycję, nie zmieniaj wartości bitu sX[7]	$sX \leftarrow \{sX[7], sX[7:1]\}$ $CARRY \leftarrow sX[0]$	?	?
STORE sX, (sY) (STORE sX, sY)	Zapisz zawartość pamięci podręcznej RAM, wskazywanej przez rejestr sY, wartością rejestru sX	$RAM[(sY)] \leftarrow sX$	-	-
STORE sX, ss	Zapisz zawartość pamięci podręcznej RAM, wskazywanej przez wartość ss, wartością rejestru sX	$RAM[ss] \leftarrow sX$	-	-
SUB sX, kk	Różnica rejestru sX i stałej kk	$sX \leftarrow sX - kk$	?	?
SUB sX, sY	Różnica rejestru sX i rejestru sY	$sX \leftarrow sX - sY$	?	?
SUBCY sX, kk (SUBC)	Różnica rejestru sX i stałej kk z bitem przeniesienia CARRY	$sX \leftarrow sX - kk - CARRY$	?	?
SUBCY sX, sY (SUBC)	Różnica rejestru sX i rejestru sY z bitem przeniesienia CARRY	$sX \leftarrow sX - sY - CARRY$	?	?
TEST sX, kk	Test bitów rejestru sX z wartością kk	Jeśli $(sX \text{ AND } kk) = 0$ , $ZERO \leftarrow 1$ $CARRY \leftarrow$ parzystość bitów wyrażenia $(sX \text{ AND } kk)$	?	?
TEST sX, sY	Test bitów rejestru sX z wartością rejestru sY	Jeśli $(sX \text{ AND } sY) = 0$ , $ZERO \leftarrow 1$ $CARRY \leftarrow$ parzystość bitów wyrażenia $(sX \text{ AND } sY)$	?	?
XOR sX, kk	Logiczna suma XOR bitów rejestru sX i stałej kk	$sX \leftarrow sX \text{ XOR } kk$	?	0
XOR sX, sY	Logiczna suma XOR bitów rejestru sX i rejestru sY	$sX \leftarrow sX \text{ XOR } sY$	?	0

sX – jeden z 16 rejestrów: od s0 do sF  
sY – jeden z 16 rejestrów: od s0 do sF  
aaa – 10-bitowy adres, możliwe są tylko wartości heksadecymalne od 000 do 3FF (od 0 do 1023)  
kk – 8-bitowa stała w postaci heksadecymalnej: od 00 do FF  
pp – 8-bitowy adres portu w postaci heksadecymalnej: od 00 do FF  
ss – 6-bitowy adres pamięci podręcznej w postaci heksadecymalnej: od 00 do 3F  
RAM[n] – zawartość podręcznej pamięci pod adresem n  
TOS – wartość na szczycie stosu

nywania warunkowych skoków programu. Instrukcja JUMP powoduje bezwarunkową zmianę wykonywania kodu od pozycji wskazanej instrukcją. Licznik programu (*Program Counter* – PC) jest 10-bitowy i może przyjmować wartości od 0 do 1023. Jeśli PC osiągnie maksymalną wartość, nastę-

na instrukcja będzie wykonana spod adresu 0 (o ile nie był to skok).

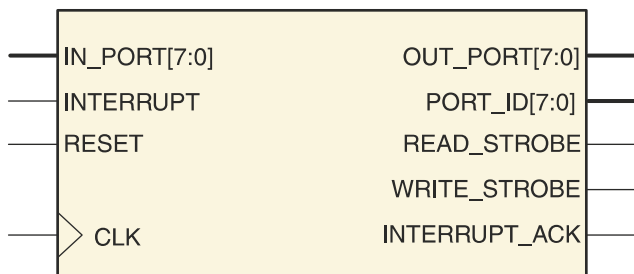
PicoBlaze może wykonywać podprogramy. Instrukcja CALL powoduje skok do podprogramu, a RETURN powrót z wykonania programu. Dzięki pamięci stosu, można zagnieździć wykonywanie kodu maksymalnie do 31 podprogramów. Jest to w zupełności wystarczające dla ograniczonego rozmiaru pamięci programu. Trzeba jednak pamiętać, że dla podprogramu przerwania jest zarezerwowany jeden poziom.

Wszystkie instrukcje wymagają dwóch

cykli zegarowych na ich wykonanie. Stałość tego parametru jest dość istotna dla potrzeb określenia sposobu działania urządzenia. Zostało to przygotowane tak, aby uzyskać w miarę wysokie częstotliwości działania PicoBlaze w strukturach FPGA.

**Implementacja w strukturach FPGA**

PicoBlaze po implementacji w układzie Spartan-3 zajmuje zaledwie 96 slice'ów i jeden blok pamięci. Dla porównania, małej pojemności układ XC3S200, używany na płytce referencyjnej ZL10PLD, ma pojemność 1920 slice'ów. Przy takiej implementacji, mikroprocesor może być taktowany sygnałem zegarowym do 88 MHz (44 MIPS – 44 miliony instrukcji wykonanych w ciągu jednej sekundy). W szybszych układach, np. Virtex-II Pro sygnał zegarowy może osiągnąć częstotliwość do 200 MHz (100 MIPS).



Rys. 5. Symbol graficzny PicoBlaze z liniami I/O



Tab. 3. Zestawienie sygnałów wejściowych i wyjściowych PicoBlaze

Nazwa sygnału	Kierunek	Szerokość	Opis
instruction	wejście	18	Aktualnie wykonywana instrukcja
address	wyjście	10	Określa adres pamięci programu, z którego jest pobierana instrukcja
in_port	wejście	8	Port wejściowy
out_port	wyjście	8	Port wyjściowy
port_id	wyjście	8	Numer aktualnie wybranego portu
read_strobe	wyjście	1	Sygnal aktywny w stanie wysokim, gdy następuje odczyt wartości portu wejściowego
write_strobe	wyjście	1	Sygnal aktywny w stanie wysokim, gdy następuje zapis do portu wyjściowego
interrupt	wejście	1	Wejście zewnętrznego przerwania, sygnał ten powinien być zsynchronizowany z sygnałem zegarowym, aktywny stan wysoki
interrupt_ack	wyjście	1	Potwierdzenie przyjęcia przerwania, aktywny stan wysoki
reset	wejście	1	Sygnal resetujący mikroprocesor, aktywny stan wysoki
clk	wejście	1	Sygnal taktujący (zegarowy)

Wbudowana w rdzeń PicoBlaze podręczna pamięć RAM (*scratchpad RAM*) ma pojemność 64 bajtów. Dostęp do niej jest realizowany przez instrukcje FETCH i STORE. Pozwalają one na wymianę danych pomiędzy rejestrami a podręczną pamięcią, wykorzystując adresowanie pośrednie i bezpośrednie. Dzięki podręcznej pamięci RAM, w większości zastosowań

PicoBlaze, sygnały wejścia-wyjścia można w całości przeznaczyć na operacje rozszerzające możliwości procesora, a nie na obsługę dodatkowej pamięci RAM.

PicoBlaze wyposażono ponadto w 256 portów wejściowych i 256 portów wyjściowych. Każdy port jest adresowany ośmiobitową magistralą PORT\_ID. Operacje wyjściowe są

rozróżniane impulsem sygnału WRITE\_STROBE. Operacje wejściowe są rozróżniane impulsem sygnału READ\_STROBE. Dzięki bardzo uproszczonej budowie, porty te pozwalają podłączyć do mikroprocesora dowolne periferia, jak i dodatkową pamięć RAM.

Porty mogą być adresowane bezpośrednio przez instrukcję lub pośrednio przez rejestr. Dostęp do portu realizowany jest za pomocą instrukcji: INPUT i OUTPUT.

PicoBlaze obsługuje jedno przerwanie zewnętrzne. Powinno ono być zsynchronizowane z sygnałem zegarowym mikroprocesora. Obsługa przerwania jest bardzo szybka i wymaga jedynie 5 taktów sygnału zegarowego.

Zerowanie PicoBlaze'a następuje automatycznie po załadowaniu konfiguracji układu FPGA. Licznik programu PC jest ustawiony na 0, znaczniki są wyzerowane, przerwania zablokowane i pamięć stosu wyzerowana.

W tab. 3 przedstawiono opis wejść i wyjść mikroprocesora PicoBlaze, a na rys. 5 symbol blokowy mikroprocesora.

**Marcin Nowakowski**

**WYPOSAŻENIE SERWISANTA**

**VTTEST5**  
Próbnik napięcia  
15.00 zł

**VTBAG2**  
Torba-kuferek na narzędzia  
44.00 zł

**DVM760**  
Miernik uniwersalny z testerem sieci LAN  
57.00 zł

**VTBAG1**  
Torba na narzędzia  
42.00 zł

**VTSET18**  
Zestaw narzędzi 11 szt.  
43.00 zł

[www.sklep.avt.pl](http://www.sklep.avt.pl) • tel. 022 257 84 50

**velleman<sup>®</sup> INSTRUMENTS**

**ZESTAW LABORATORYJNY LAB-1**

**3w1**

**Multimetr cyfrowy:**

- \* podświetlany wyświetlacz LCD 3 1/2"
- \* automatyczny wskaźnik polaryzacji
- \* zakres pomiarowy napięcia stałego: od 200 mV do 600 V (5 podzakresów)
- \* zakres pomiarowy napięcia zmiennego: od 200 V do 600 V
- \* zakres pomiarowy prądu stałego: od 200 µA do 10 A (5 podzakresów)
- \* pomiar rezystancji: od 200 Ω do 2 MΩ
- \* tester diod, tranzystorów i zwarcia
- \* funkcja HOLD
- \* sygnalizacja dźwiękowa

**Zasilacz:**

- \* napięcia wyjściowe: 3 - 4,5 - 6 - 7,5 - 9 - 12 VDC
- \* mały poziom tętnień
- \* maksymalny prąd obciążenia: 1,5 A (2 A szczytowo)
- \* wskaźnik przeciążenia - dioda LED
- \* wskaźnik załączenia - dioda LED

**Stacja lutownicza:**

- \* zasilanie lutownicy - 24 V
- \* grzałka - ceramiczna, o mocy 48 W, wbudowany czujnik temperatury
- \* zakres temperatur grota 150 - 450°C
- \* możliwość lutowania bezolowowego
- \* w zestawie gąbka czyszcząca

**Cena brutto: 379 zł**

Zamówienia przyjmuje Dział Handlowy AVT  
03-197 Warszawa, ul. Leszczynowa 11  
tel. 022 257 84 50, fax 022 257 84 55  
e-mail: [handlowy@avt.pl](mailto:handlowy@avt.pl), [www.sklep.avt.pl](http://www.sklep.avt.pl)