

Zmodyfikowane mnożenie o stałej szerokości bitowej

Ernest Jamro, Maciej Wielgosz, Paweł Russek, Kazimierz Wiatr

Streszczenie: Niniejszy artykuł prezentuje nową metodę kompensacji błędu odcięcia dla mnożenia o stałej szerokości bitowej czyli takiej, dla której szerokość bitowa argumentów wejściowych jest taka sama jak wyjścia. Niektóre poprzednie publikacje były oparte na błędnych założeniach, dlatego zadaniem tej publikacji jest wykazanie wspomnianych błędów oraz zaprezentowanie nowej architektury, dla której błąd średni dąży do zera.

Słowa kluczowe: arytmetyka cyfrowa, filtry cyfrowe, układ mnożący

1. WPROWADZENIE

Operacja mnożenia jest jedną z podstawowych operacji arytmetycznych i dlatego liczba zasobów sprzętowych zajmowanych przez układy mnożące ma bardzo duże znaczenie. Operacja mnożenia jest zazwyczaj wykonywana z pełną szerokością bitową, to jest taką, dla której liczba bitów, które wykorzystywane są do reprezentacji wyniku mnożenia jest równa sumie szerokości bitowej argumentów wejściowych. Warto podkreślić, że w większości aplikacji szerokości danych wejściowych i wyjściowych są takie same lub różnią się nieznacznie. Przykładem może być np. przetwarzanie sygnałów cyfrowych.

W takim wypadku możliwe jest zredukowanie zasobów układu mnożącego poprzez odcięcie młodszych bitów oraz zastosowanie dodatkowej logiki kompensacji błędu [2, 3].

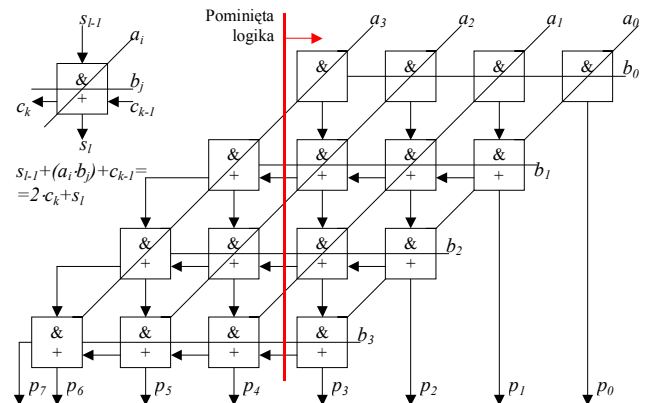
Główną ideą niniejszego artykułu jest wykazanie błędnych założeń niektórych poprzednich publikacji dotyczących tego tematu oraz zaproponowanie nowego rozwiązania, które charakteryzuje się mniejszym błędem średnim w stosunku do poprzednich rozwiązań.

2. MNOŻENIE O SKRÓCONEJ SZEROKOŚCI

Jedną z podstawowych metod implementacji operacji mnożenia jest równoległy układ mnożący (ang. parallel-array multiplier) [1], którego schemat blokowy został przedstawiony na rys. 1. Warto zauważyć, że nie jest to optymalna metoda implementacji wspomnianej operacji, jednak ze względu na swoją prostotę to właśnie ona będzie tutaj rozpatrywana podobnie jak w [2, 3].

Istnieje wiele metod użycia takiego układu mnożącego w przypadku, gdy szerokość bitowa argumentów wejściowych n jest taka sama jak szerokość bitowa wyniku. Standardowa metoda polega na implementacji całego układu mnożącego przedstawionego na rys. 1, dla którego szerokość bitowa wyjścia wynosi $2n$ (jest dwa razy większa niż szerokość bitowa wejść n). Następnie szerokość bitowa wyjścia jest ograniczana poprzez odrzucenie n najmłodszych bitów wyniku;

operacji tej zwykle towarzyszy operacja zaokrąglania. Metoda ta będzie dalej nazywana pełnym mnożeniem i jest ona najdokładniejsza, jeśli chodzi o błąd obliczeń i zarazem najgorsza, jeśli chodzi o ilość zajmowanych zasobów.



Rys. 1. Układ mnożący równoległy dla $n=4$

Inną metodą implementacji operacji mnożenia ze stałą szerokością jest struktura z bezpośrednim odcięciem najmłodszych n -bitów przedstawiona na rys. 1, dla której cała logika na prawo od bitu p_n nie jest implementowana. Struktura ta charakteryzuje się najmniejszą zajmowaną powierzchnią z jednej strony, ale największym błędem obliczeń z drugiej strony. Analizując rys. 1 można zauważyć, że struktura ta zajmuje niewiele mniej niż połowę zasobów pełnego układu mnożącego.

Rozważając dwie dane wejściowe n -bitowe bez znaku:

$$A = \sum_{i=0}^{n-1} a_i 2^i \quad B = \sum_{i=0}^{n-1} b_i 2^i,$$

gdzie $a_i, b_i \in \{0,1\}$; wynik pełnego mnożenia P_{std} można przedstawić jako:

$$P_{std} = A \cdot B = 2^{-n} \cdot \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} a_i \cdot b_j \cdot 2^{i+j} \quad (1)$$

W powyższym wzorze faktyczny wynik mnożenia jest skalowany przez 2^{-n} tak, aby rozpatrywany wynik P_{std} był w podobnym zakresie liczbowym co dane wejściowe A, B . Wynik mnożenia P_{std} można podzielić na dwie części:

$$P_{std} = MP + LP \quad (2)$$

$$MP = 2^{-n} \sum_{i+j \geq n} (a_i b_j \cdot 2^{i+j}) \quad (3)$$

$$LP = 2^{-n} \sum_{i+j < n} (a_i b_j \cdot 2^{i+j}) \quad (4)$$

gdzie: MP oznacza bardziej znaczące bity wyniku, które są zawsze implementowane (na lewo od linii odcięcia na rys.

1); LP oznacza młodsze bity wyniku, które są odrzucane (na prawo od linii odcięcia na rys. 1). W konsekwencji dla układu mnożącego bez dodatkowej logiki kompensacji błędu odcięcia, błąd ten wynosi LP .

W celu obliczenia błędu odcięcia założono, że prawdopodobieństwo wystąpienia bitu 0 i 1 dla każdego bitu wejściowego a_i , b_j jest równe i wynosi 0.5. W konsekwencji wartość oczekiwana iloczynu logicznego $E\{a_i b_j\} = 0.25$.

Analizując rys. 1 oraz (4), można zauważyć, że wartość oczekiwana $E\{LP\}$ wynosi [4]:

$$E\{LP\} = E\{a_i b_j\} \cdot 2^{-n} \sum_{i+j < n} 2^{i+j} = 0.25 \cdot 2^{-n} \sum_{i=1}^n i \cdot 2^{i-1} \quad (5)$$

Błąd odcięcia może być zredukowany poprzez dodanie wartości stałej równej $E\{LP\}$ zaokrąglonej do najbliższej liczby całkowitej [5]. Innym rozwiązaniem jest zastosowanie zmiennej kompensacji błędu oznaczanej jako Variable Correction Truncated Multiplier (VCTM) [2]. Dla tej kompensacji błędu wyrażenie LP zastało podzielony na dwie składowe:

$$LP = LP_{main} + LP_{remain} \quad (6)$$

$$LP_{main} = \frac{1}{2} \sum_{i+j=n-1} a_i b_j = \frac{1}{2} (a_{i-1} b_0 + a_{i-2} b_1 + \dots + a_0 b_{i-1}) \quad (7)$$

$$LP_{remain} = \frac{1}{4} \sum_{i+j=n-2} a_i b_j + \frac{1}{8} \sum_{i+j=n-3} a_i b_j + \dots + \frac{1}{2^{n-1}} (a_0 b_1 + a_1 b_0) + \frac{1}{2^n} a_0 b_0 \quad (8)$$

Warto zauważyć, że wartość LP jest najbardziej zależna od wartości LP_{main} , ponieważ współczynnik skalujący wynosi $\frac{1}{2}$ w porównaniu z LP_{remain} , dla której współczynnik mnożenia wynosi $\frac{1}{4}$ lub więcej. Dlatego dla VCTM, do kompensacji błędu odcięcia używany jest tylko czynnik LP_{main} . Aby rekompensować odrzucenie wyrażenia LP_{remain} , czynnik LP_{main} jest brany z wagą dwa razy większą:

$$P_{VCTM} = MP + 2 \cdot LP_{main} = MP + \beta \quad (9)$$

gdzie:

$$\beta = 2 \cdot LP_{main} = a_{n-1} b_0 + a_{n-2} b_1 + \dots + a_0 b_{n-1}. \quad (10)$$

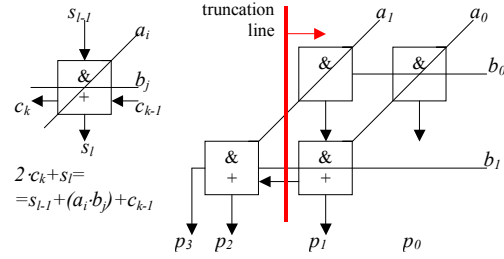
Inne metody kompensacji błędu odcięcia zostały zaproponowane przez Jou et. al. [3]. Niestety są one oparte na szeregu złych założeń.

Pierwsze z tych założeń polega na tym, że nie jest konieczne rozpatrywanie wszystkich kombinacji danych wejściowych a jedynie wystarczy rozpatrywać osobno iloczyny bitowe $a_i b_j$ oraz wartość oczekiwaną jego wystąpienia $E\{a_i b_j\}$. Jest to wyraźnie stwierdzone w zdaniu [4]: Jako że każdy iloczyn bitowy $a_i b_j$ może być uważany za niezależną zmienną, wariancja sumy iloczynów $a_i b_j$ jest równa sumie wariancji $a_i b_j$.

$$\sigma_{[4]} = \sigma_{AND} \cdot 2^{-t} \sqrt{\sum_{i=1}^t i \cdot 2^{2(i-1)}} \quad (11)$$

gdzie $\sigma_{AND}^2 = \frac{3}{16}$ jest wariancją iloczynu bitowego $a_i b_j$.

Powyższe założenie jest podane bez żadnych dodatkowych rozważań. Aby udowodnić, że powyższe założenie jest błędne rozważmy prosty układ mnożący z bezpośrednim odcięciem dla $n=2$, który jest podany na rys. 2.



Rys. 2. Układ mnożący z bezpośrednim odcięciem dla $n=2$.

Tab. 1. ilustruje zachowanie tego układu dla wszystkich kombinacji danych wejściowych. Warto zauważyć, że obliczona wariancja σ^2 różni się od wariancji $\sigma_{[4]}^2$ obliczonej według (11). Co potwierdza nieprawidłowość rozważanego założenia. Powodem tego, że (11) nie jest prawidłowe jest fakt, że iloczyny bitowe są ze sobą powiązane. Na przykład iloczyn $b_1 a_0$ oraz $b_0 a_0$ mają wspólną daną wejściową a_0 . Warto zwrócić uwagę, że rozważane założenie jest prawdziwe dla równań liniowych, niestety równanie (11) nim nie jest.

Tab. 1. Różne błędny odcięcia dla wszystkich danych wejściowych

i	b_1	b_0	a_1	a_0	$b_1 a_0$	$b_0 a_1$	$a_0 b_0$	LP_i	β (10)	γ (17)	α^2 (12)
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0	0
3	0	0	1	1	0	0	0	0	0	0	0
4	0	1	0	0	0	0	0	0	0	0	0
5	0	1	0	1	0	0	1	0.25	0	0	0
6	0	1	1	0	0	1	0	0.5	1	0	0
7	0	1	1	1	0	1	1	0.75	1	0	0
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	1	0	0	0.5	1	1	0
10	1	0	1	0	0	0	0	0	0	0	0
11	1	0	1	1	1	0	0	0.5	1	1	0
12	1	1	0	0	0	0	0	0	0	0	0
13	1	1	0	1	1	0	1	0.75	1	1	0
14	1	1	1	0	0	1	0	0.5	1	0	0
15	1	1	1	1	1	1	1	1.25	2	1	1
$E\{LP_i\}$	0.5	0.5	0.5	0.5	0.25	0.25	0.25	0.3125	0.5	0.25	0.0625
$E\{LP_i\} = \frac{1}{16} \sum_{i=0}^{15} LP_i = \frac{1}{16} \sum_{i=0}^{15} [0.5 \cdot (b_1 a_0 + b_0 a_1) + 0.25 \cdot a_0 b_0]$											0.3125
$\sigma = \sqrt{E\{(LP_i - E\{LP_i\})^2\}} = \sqrt{\frac{1}{16} \sum_{i=0}^{15} (LP_i - 0.3125)^2}$											0.3698
$\sigma_{[4]} = \sigma_{AND} \cdot 2^{-n} \cdot \sqrt{\sum_{i=1}^{n-w} i \cdot 2^{2(i-1)}} = \frac{\sqrt{3}}{4} \cdot \sqrt{\sum_{i=1}^2 i \cdot 2^{2(i-1)}}$											0.3248

Warto podkreślić, że powyższe fałszywe założenie jest najprawdopodobniej używane również w innych publikacjach: [3, 6, 7]. Wskazują na to dowód twierdzenia 1 [3], (25) oraz (26) [6], (27) oraz (30) [7]. Warto podkreślić, że powyższe publikacje używają nieliniowej kompensacji błędu, np. (9) [3] używa:

$$\alpha' = \begin{cases} 0, & \text{if } \beta = 0 \\ \beta - 1 & \text{if } \beta > 0 \end{cases} \quad (12)$$

Tab. 1. pokazuje nieprawidłowość działania równania (12). Wartość oczekiwana idealnej kompensacji błędu wynosi $E\{LP_i\} = 0.3125$, natomiast wartość oczekiwana wspomnianej kompensacji błędu $E\{\alpha_i\} = 0.0625$.

Innym niepoprawnym założeniem jest uznanie, że wartość średnia jest równa średniej arytmetycznej wartości minimalnej i maksymalnej. Złożenie to zostało użyte w [3] podczas udowadniania lematu 2. Warto wspomnieć, że lemat 2 jest twierdzeniem pomocniczym równania (12), na podstawie, którego zbudowany jest układ mnożący.

3. NOWA KOMPENSACJA BŁĘDU ODCIĘCIA

Celem niniejszego artykułu jest przedstawienie nowej, poprawnej metody kompensacji błędu odcięcia. Jest ona wzorowana na VCTM [2] i dlatego zaproponowany układ będzie nazywany jako Improved Variable error Compensation Truncated Multiplier (IVCTM).

Błąd średni, będący podstawą zaproponowanego układu, jest obliczany według równania:

$$ME = \frac{1}{N} \sum_{i=1}^N e_i \quad (13)$$

gdzie e_i jest różnicą pomiędzy wartością poprawną (wynik dla pełnego układu mnożącego) a wartością otrzymaną dla i -tej próbki zredukowanego układu mnożącego; N - jest ilością rozpatrywanych różnych kombinacji danych wejściowych A , B , i jest ona ograniczona od góry przez $N = 2^{2n}$.

Dla układu mnożącego VCTM wartość błędu średniego może być obliczona według wzoru:

$$\begin{aligned} ME_{VCTM} &= E\{LP - \beta\} = 0.25 \cdot [2^{-n} \sum_{i=1}^n (i \cdot 2^{i-1}) - n] = \\ &= 0.25 \cdot [2^{-n} \sum_{i=1}^{n-1} (i \cdot 2^{i-1}) - \frac{n}{2}] \end{aligned} \quad (14)$$

Suma w (14) może być uproszczona według matematycznej tożsamości:

$$\begin{aligned} \sum_{i=1}^{n-1} (i \cdot 2^{i-1}) &= \sum_{i=1}^{n-1} \sum_{j=i}^{n-1} 2^{j-1} = \frac{1}{2} \sum_{i=1}^{n-1} (2^n - 2^i) = \frac{1}{2} [(n-1) \cdot 2^n - \sum_{i=1}^{n-1} 2^i] = \\ &= \frac{1}{2} [(n-1) \cdot 2^n - 2^n + 2] = \frac{(n-2) \cdot 2^n}{2} + 1 \end{aligned} \quad (15)$$

W konsekwencji błąd średni ME_{VCTM} wynosi:

$$ME_{VCTM} = 0.25 \cdot [2^{-n} \cdot (\frac{(n-2) \cdot 2^n}{2} + 1) - \frac{n}{2}] = 0.25 \cdot [2^{-n} - 1] \quad (16)$$

Z równania (16) można wywnioskować, że dla relatywnie dużych n błąd $ME_{VCTM} \approx 0.25$. Ponieważ wartość oczekiwana iloczynu bitowego $E\{a_i b_j\} = 0.25$, możliwe jest ulepszenie układu VCTM poprzez usunięcie jednej bramki AND; w konsekwencji wartość $ME_{IVCTM} \approx 0$. Kompensacja błędu odcięcia dla zaproponowanego układu mnożącego IVCTM powinna wyglądać następująco:

$$\gamma = \beta - a_{n-1}b_0 = a_{n-2}b_1 + a_{n-2}b_2 + \dots + a_0b_{n-1} \quad (17)$$

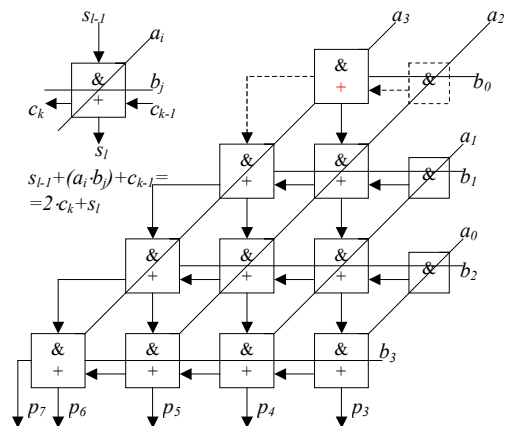
Wartość średnia błędu dla zaproponowanego układu ME_{IVCTM} jest o 0.25 mniejsza niż dla VCTM i na podstawie (16) wynosi:

$$ME_{IVCTM} = ME_{VCTM} - 0.25 = \frac{1}{4 \cdot 2^n} \quad (18)$$

Wartość ta szybko dąży do zera dla wzrastających szerokości bitowych n .

Schemat układu IVCTM dla $n=4$ został przedstawiony na rys. 3. Dla porównania układ VCTM wymaga dodatkowej bramki AND która została zaznaczona na rys. 3 przerywaną linią. Zaproponowana kompensacja błędu odcięcia γ składa się z dwóch bramek AND: a_1b_1 oraz a_0b_2 . Dla układu VCTM wymagana jest dodatkowa bramka AND: a_2b_0 .

Warto podkreślić, że na pierwszy rzut oka zaproponowany układ IVCTM w porównaniu z VCTM wymaga użycia tylko jednej bramki AND mniej. Dokładna analiza pokazuje jednak, że liczba zajmowanych zasobów sprzętowych jest znacznie mniejsza, ponieważ pewne układy sumatorów ulegają redukcji. W rozważanym przypadku na rys. 3, półsumator iloczynu cząstkowego a_3b_0 został zastąpiony zwykłą bramką AND. Podobnie pełny sumator iloczynu cząstkowego a_3b_1 został zastąpiony półsumatorem.



Rys. 3. Schemat układu IVCTM dla $n=4$

4. WYNIKI IMPLEMENTACJI

Analizując wartości błędów dla VCTM oraz IVCTM można zauważyć, że ME zamieszczony w tab. 2 jest zdecydowanie mniejszy dla zaproponowanego układu IVCTM. Natomiast błąd RMSE zamieszczony w tab. 3 dla układów VCTM i IVCTM jest niemal identyczny.

RMSE (Root Mean Square Error) - błąd średniokwadratowy jest obliczany według poniższego wzoru:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N e_i^2} \quad (19)$$

Tab. 2. Średni błąd dla różnych metod kompensacji

n	direct	Jou [3]	VCTM	IVCTM
4	0.766	0.449	-0.235	0.015
5	1.008	0.521	-0.242	0.008
6	1.254	0.576	-0.246	0.004
7	1.502	0.618	-0.248	0.002
8	1.750	0.651	-0.249	0.002
9	2.000	0.675	-0.250	0.000
10	2.250	0.694	-0.250	0.000
11	2.499	0.708	-0.250	0.001
12	2.750	0.718	-0.250	0.000
13	3.000	0.726	-0.250	0.000
14	3.251	0.731	-0.251	0.000
15	3.501	0.736	-0.250	0.000
16	3.750	0.740	-0.250	0.000

Tab. 3. RMSE dla różnych metod kompensacji błędów odcięcia

n	direct	Jou [3]	VCTM	IVCTM
4	0.984	0.559	0.416	0.416
5	1.240	0.629	0.444	0.444
6	1.495	0.686	0.470	0.470
7	1.750	0.733	0.492	0.492
8	2.004	0.774	0.513	0.513
9	2.258	0.807	0.534	0.533
10	2.511	0.837	0.553	0.553
11	2.763	0.862	0.572	0.571
12	3.016	0.883	0.589	0.590
13	3.268	0.903	0.606	0.606
14	3.521	0.920	0.624	0.624
15	3.772	0.936	0.640	0.640
16	4.023	0.952	0.656	0.656

5. WNIOSKI

Po wykazaniu, że uprzednio opublikowane układy mnożące o zredukowanej szerokości są oparte na błędnych założeniach, konieczne stało się opracowanie nowego układu. Zastosowanie autorskiej kompensacji błędów zmniejsza zdecydowanie błąd średni w porównaniu z układem VCTM [2], pozostawiając na dotychczasowym poziomie błąd RMSE. Warto podkreślić, że liczba zajmowanych zasobów przez wspomniany układ mnożący uległa zmniejszeniu w porównaniu z VCTM.

LITERATURA

- [1] A. R. Omondi, *Computer Arithmetic Systems: Algorithms, Architecture and Implementation*, Prentice-Hall International, 1994
- [2] E. J. King and E. E. Swartzlander Jr., *Data-dependent truncation scheme for parallel multipliers*, in Proc. 31st Asilomar Conf. Signals, Systems, and Computers, vol. 2, Pacific Grove, CA, pp. 1178–1182, 1997
- [3] J. M. Jou, S. R. Kuang, and R. D. Chen, *Design of low-error fixed-width multiplier for DSP applications*, IEEE Trans. Circuits Syst. II, Analog Digit. Signal. Process., vol. 46, no. 6, pp. 836–842, Jun. 1999.
- [4] Y. C. Lim, *Single-precision multiplier with reduced circuit complexity for signal processing applications*, IEEE Trans. Comput., vol. 41, no. 10, pp. 1333–1336, Oct. 1992.
- [5] M. J. Schulte, E. E. Swartzlander Jr., *Truncated multiplication with correction constant*, in VLSI Signal Processing, VI. New York: IEEE Press, pp. 388–396, 1993
- [6] L. D. Van, S. S. Wang, and W. S. Feng, *Design of the lower-error fixed-width multiplier and its application*, IEEE Trans. Circuits Syst. II, Analog Digit. Signal. Process., vol. 47, pp. 1112–1118, Oct. 2000.
- [7] L. D. Van, C. C. Yang, *Generalized Low-Error Area-Efficient Fixed-Width Multipliers*, IEEE Transactions on Circuits and Systems—I, Vol. 52, No. 8, pp. 1608–1619, Aug. 2005.



dr inż. Ernest Jamro
Akademia Górniczo-Hutnicza,
WEAiE, Kat. Elektroniki
Al. Mickiewicza 30, 30-059 Kraków

ACK Cyfronet AGH
ul. Nawojki 11, 30-950 Kraków

tel.: +48-12-6172792
e-mail: jamro@agh.edu.pl



dr inż. Maciej Wielgosz
Akademia Górniczo-Hutnicza,
WEAiE, Kat. Elektroniki
Al. Mickiewicza 30, 30-059 Kraków

ACK Cyfronet AGH
ul. Nawojki 11, 30-950 Kraków

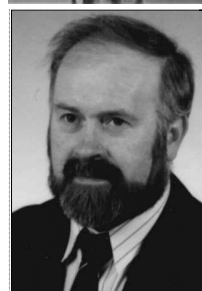
tel.: +48-12-6172792
e-mail: wielgosz@agh.edu.pl



dr inż. Paweł Russek
Akademia Górniczo-Hutnicza,
WEAiE, Kat. Elektroniki
Al. Mickiewicza 30, 30-059 Kraków

ACK Cyfronet AGH
ul. Nawojki 11, 30-950 Kraków

tel.: +48-12-6172792
e-mail: russek@agh.edu.pl



prof. dr hab. inż. Kazimierz Wiatr
Akademia Górniczo-Hutnicza,
WEAiE, Kat. Elektroniki
Al. Mickiewicza 30, 30-059 Kraków

ACK Cyfronet AGH
ul. Nawojki 11, 30-950 Kraków

tel.: +48-12-6173033
e-mail: wiatr@agh.edu.pl