

論文 2003-40SC-3-1

# DSP를 이용한 확장 가능한 디지털 펄스압축기 설계

## (The Design of Expansible Digital Pulse Compressor Using Digital Signal Processors)

申鉉翼\*, 柳榮鎭\*, 金煥宇\*\*

(Hyun-Ik Shin, Young-Jin Ryoo, and Whan-Woo Kim)

### 요약

디지털 신호처리에 대한 성능향상과 더불어 레이더 펄스 압축기 또한 디지털 처리방식이 점점 일반화 되어가고 있다. 디지털 펄스압축기는 FIR(finite impulse response) 필터 알고리즘을 이용한 시간영역 처리방식 또는 FFT(fast Fourier transform) 알고리즘을 이용한 주파수영역 처리방식으로 구현될 수 있다. 본 논문에서는 다중 DSP(digital signal processor)를 이용하여 확장성이 용이한 디지털 펄스압축기를 구성하고, 아날로그 디바이스사의 ADSP-21060을 적용하여 수신 거리 셀 및 FIR 필터 탭(tap) 수에 따른 펄스압축 연산시간을 C-언어와 어셈블리 언어로 시간영역에서 비교·분석하였다. 분석결과를 적용함으로써, 레이더의 시스템 파라미터가 정해지는 경우 펄스압축기 구성에 소요되는 DSP 수를 쉽게 예측할 수 있음을 확인하였다.

### Abstract

With the improvement of digital signal processors, digital pulse compressor(DPC) is widely used in radar systems. The DPC can be implemented by using FIR filter algorithm in time domain or FFT algorithm in frequency domain. This paper designs an expansible DPC using multiple DSPs. With ADSP-21060 of Analog Devices Inc., the computation time as a function of the number of received range cells and FIR filter tap is compared and analyzed in time domain using C-language and assembly language. Therefore, when radar system parameters are determined, the number of DSP's required to implement DPC can be easily estimated.

**Keyword** : digital pulse compression, radar, digital signal processor

### I. 서론

펄스압축(PC : pulse compression) 기법은 원하는 송

신전력을 얻기 위해 펄스 폭을 길게 하면서도 좁은 펄스 폭에 해당하는 거리분해능(resolution)을 얻기 위해 탐색(surveillance) 및 추적 레이더에 주로 사용된다. 이 기법을 사용하는 경우 송신신호에는 대역폭을 넓게 하기 위한 주파수 또는 위상변조가 가해지며, 수신신호는 정합(matched)필터에 의해 압축된다. 이러한 기능의 펄스압축 처리는 디지털 신호처리의 성능향상과 더불어 최근에는 디지털 방식에 의해 구현되며, 아날로그 방식에 비해 큰 압축 비를 쉽게 구현할 수 있는 장점이 있다<sup>[1-3]</sup>.

\* 正會員, 國防科學研究所

(Agency for Defense Development)

\*\* 正會員, 忠南大學校 情報通信工學部

(Division of Electrical and Computer Engineering,

Chungnam National University)

接受日字:2002年9月13日, 수정완료일:2003年4月22日

디지털 펄스압축 처리는 FIR 필터 알고리즘을 이용한 시간영역 또는 FFT 알고리즘을 이용한 주파수 영역에서 구현될 수 있다. FFT에 의한 주파수영역 연산은 반드시 거리 셀( $N_R$ ) 전체가 수신된 후 이를 블록(block) 단위로 처리해야 하지만, FIR 필터 알고리즘에 의한 시간영역 연산은 하드웨어 구성에 따라서는 거리 셀 데이터를 수신하고 있는 중에도 필터연산을 시작할 수 있는 유리한 점이 있다<sup>[2-4]</sup>.

본 논문에서는 다중 DSP를 이용하여 확장성이 용이한 디지털 펄스압축기를 구성하였다. 아날로그 디바이스사의 ADSP-21060을 사용하여 C-언어와 어셈블리 언어에 의한 시간영역 펄스압축 연산시간을 분석함으로써, 시스템 파라미터가 정해지는 경우 다중 DSP 구조의 펄스압축 처리기 구성에 소요되는 DSP 수를 쉽게 예측할 수 있도록 하였다. 그리고 적용 가능성을 단거리 및 중거리 탐색 레이더를 대상으로 확인하였다.

### II. 시간영역 펄스압축

레이더의 송신 파형을  $s(i)$ ,  $i=0,1,\dots,N_T-1$  와 같이 표현하는 경우, 이 신호에 대한 정합필터의 임펄스(impulse) 응답  $h(i)$ 는

$$h(i) = s^*(N_T - i - 1), \quad i = 0, 1, 2, \dots, N_T - 1 \quad (1)$$

과 같이 주어진다. 식에서 \*는 복소공액(complex conjugate)을 의미하며,  $N_T$ 는 송신 파형에 대한 디지털 샘플 수이다.

시간영역에서의 펄스압축 관계는 수신신호  $x(i)$ 와 정합필터의 임펄스 응답  $h(i)$ 의 상승적(convolution) 임으로 식 (2)와 같이 표현될 수 있다<sup>[3]</sup>.

$$\begin{aligned} y(k) &= \sum_{i=0}^{N_T-1} x(k+i) h(N_T-1-i) \\ &= \sum_{i=0}^{N_T-1} x(k+i) s^*(-i), \\ k &= 0, 1, 2, \dots, N_R - N_T. \end{aligned} \quad (2)$$

식에서  $y(k)$ 는 펄스압축기의  $k$ 번째 출력신호,  $x(i)$ 는  $i$ 번째 수신신호,  $h(i)$ 는  $i$ 번째 임펄스 응답,  $N_T$ 은 FIR 필터 샘플 수,  $N_R$ 은 수신신호의 거리 셀 수를 각각 의미한다.

### III. 디지털 펄스압축기 구성

하드웨어 측면에서 디지털 펄스압축기 구현방식은 크게 DSP를 이용하는 방식과 FIR 필터 또는 ASIC(application specific integrated circuit)과 같은 전용기능의 칩(chip)을 이용하는 2가지로 구분될 수 있으며, DSP를 이용하는 경우에는 일반적으로 연산수행에 앞서 입력데이터를 원하는 메모리로 이동시켜야 한다<sup>[2, 5]</sup>. 따라서 펄스압축처리의 데이터 저장/분배/연산 타이밍(timing)은 <그림 1>과 같이 일반화 할 수 있다. <그림 1>에 나타낸 데이터 저장 및 분배시간은 하드웨어 구조에 따라서는 무시될 수 있으며, 이 경우에는 연산 수행이 데이터 저장과 거의 동시에 시작 될 수 있다.

DSP를 이용하여 펄스압축 처리기를 구현하는 경우, <그림 1>의 타이밍 관계에서 데이터 분배 및 펄스압축 연산시간은 일반적으로 PRI(pulse repetition interval)보다 크다. 결국 여러 개의 DSP를 이용하여야만 연산지연을 갖는 실시간 펄스압축기의 구현이 가능하며, 이를 구현하기 위하여 <그림 2>와 같이 N개의 DSP를 이용한 다중 DSP 구조의 디지털 펄스압축기를 구성하였다.

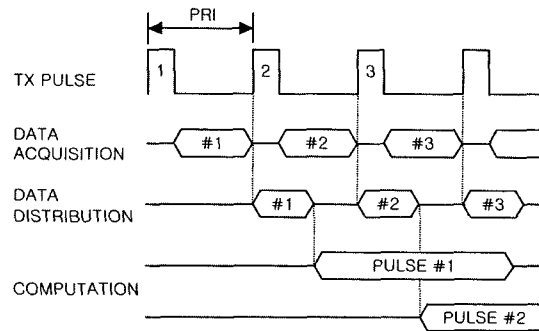


그림 1. 디지털 펄스압축기 연산 타이밍  
Fig. 1. Computation timing of DPC.

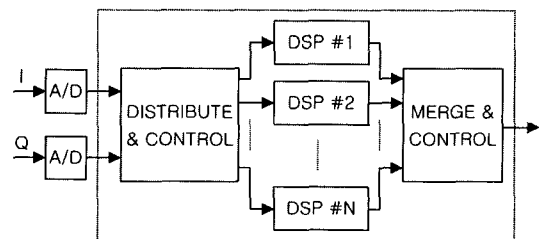


그림 2. 디지털 펄스압축기의 구성  
Fig. 2. Configuration of DPC.

<그림 2>에서 데이터 분배 및 컨트롤 회로가 A/D 보드로부터 입력되는 데이터를 첫 번째 DSP에서부터 순환(circulation)하여 분배한다면, 데이터 분배 및 펄스 압축에 소요되는 연산시간이 정해지는 경우 시스템 구성에 필요한 DSP 수를 쉽게 예측할 수 있다. 즉, 시스템 구성에 소요되는 DSP 수  $N_{DSP}$  는

$$N_{DSP} = \frac{T_D + T_{PC}}{PRI} \quad (3)$$

가 된다. 식에서  $T_D$ 는 데이터 저장 후 DSP 메모리로의 데이터 분배시간,  $T_{PC}$ 는 펄스압축 연산시간, PRI (pulse repetition interval)는 펄스반복시간을 나타낸다. 일반적으로 MTI(moving target indicator) 탐색 레이더에 있어서는  $T_{PC} \gg T_D$ 의 관계가 성립하므로 식 (3)은  $N_{DSP} \approx T_{PC}/PRI$ 와 같이 근사 화하여 사용될 수 있다.

#### IV. 펄스압축 연산시간

DSP를 이용하여 시스템이 요구하는 고속연산을 수행하는 경우, 운용 프로그램의 일반적인 제어(control) 부분은 C-언어와 같은 상위(high level) 언어로 개발되지만, 속도가 요구되는 주요 알고리즘은 어셈블리(assembly) 언어를 사용하여 개발하는 것이 일반적이다<sup>[6]</sup>. 본 절에서는 C-언어와 ADSP-21060 어셈블리 언어에 대한 펄스압축 연산속도를 비교·분석하고, 연산시간에 의한 지연을 줄일 수 있는 데이터 중첩처리 방식을 제안하였다.

본 논문에서 사용한 아날로그 디바이스 사의 ADSP-21060은 40MHz 클럭으로 32-비트 부동소수점 연산을 수행하며, 프로그램 메모리 및 데이터 메모리에서 두 개의 데이터를 동시에 접근할 수 있기 때문에 레이더와 같이 복소수(I 및 Q 채널 데이터) 연산을 수행하는 분야에 매우 적합한 특성을 갖는다. 연산 능력은 한 사이클에 3개까지의 다중연산을 수행할 수 있으므로 최대 120 MFLOPS 의 부동소수점 연산을 수행할 수 있으며, 프로세서간 데이터 전송을 위한 6개의 링크 포트를 가지고 있다<sup>[7]</sup>.

프로그램 개발 및 디버깅은 아날로그 디바이스 사의 ADI VisualDSP 4.1을 사용하였다. <그림 3>은 C-언어를 사용한 펄스압축 연산 함수(function) 프로그램의

```

/*-----*/
real[] : real input data
imag[] : imaginary input data
coef_r[] : real coefficient
coef_i[] : imaginary coefficient
RANGE : received range cell number
TAP : pulse compressor tap
/*-----*/
void compress()
{
float tmp_r,tmp_i;
int ir,i,index;

for(ir=0;ir<(RANGE-TAP+1);ir++)
{
tmp_r=tmp_i=0.0;
for(i=0;i<TAP;i++)
{
index=ir+i;
tmp_r+=real[index]*coef_r[i]-imag[index]*coef_i[i];
tmp_i+=real[index]*coef_i[i]+imag[index]*coef_r[i];
}
real[ir]=tmp_r;
imag[ir]=tmp_i;
}
}
    
```

그림 3. C-언어에 의한 펄스압축 프로그램  
Fig. 3. Pulse compression program using C-language.

예이며, <그림 4>는 어셈블리 언어를 사용한 경우 이에 대한 흐름도(flow chart)이다. 펄터 계수 값 및 데이터는 한 사이클에서 2개의 데이터를 동시에 읽어올 수 있도록 프로그램 메모리 및 데이터 메모리에 각각 분리하여 저장하였다.

<표 1>은 식 (2)의 펄스압축 연산수행을 위해, C-언어로 작성된 운용 프로그램에서 <그림 3>과 <그림 4>의 함수를 호출하는 경우 거리 셀 및 FIR 필터 탭 수에 따른 소요 클럭 수를 나타내고 있으며, C-언어를 사용한 경우에는 최적화 선택사항을 적용하였다. <표 1>의 결과에서 첫 번째 항 52 및 26은 전체 클럭 수에 비해 매우 작기 때문에 C-언어 대 어셈블리 언어의 연산속도 비  $R_T$ 는

$$R_T \approx \frac{9 + 12 * N_T}{11 + 4 * (N_T - 1)} \quad (4)$$

와 같이 표현될 수 있다. <그림 5>는 식 (4)의 연산속도 비를 나타낸 결과로서 펄스압축 탭 수가 증가함에 따라 약 3배의 연산속도 차이를 확인할 수 있다.

표 1. 펄스압축 소요 클럭 수  
Table 1. Cycles required to compute PC.

언어	소요 사이클 수 (25ns/cycle)
C	$52 + (N_R - N_T + 1) * (9 + 12 * N_T)$
어셈블리	$26 + (N_R - N_T + 1) * \{11 + 4 * (N_T - 1)\}$

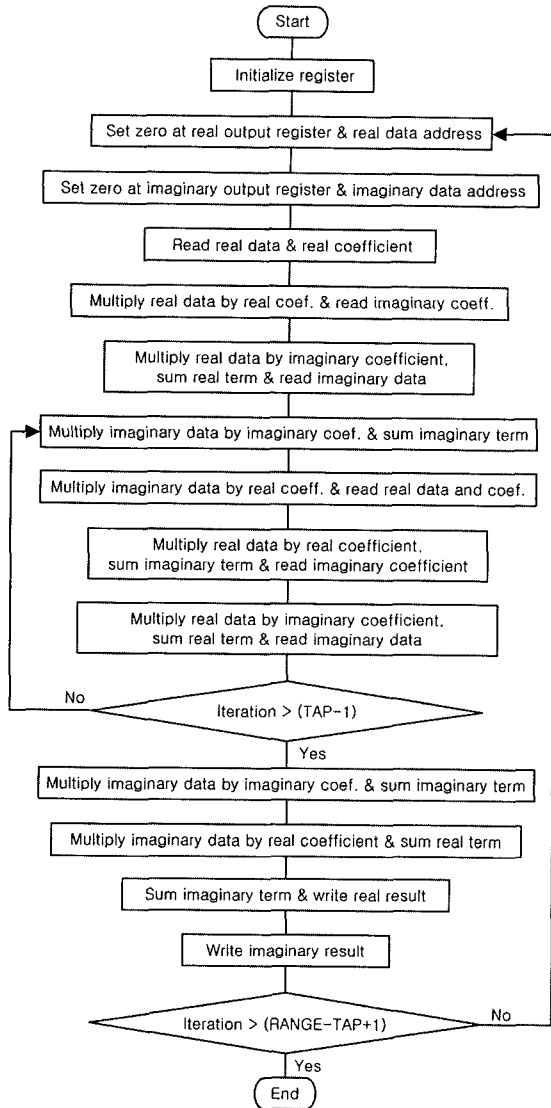


그림 4. 어셈블리어에 의한 펄스압축 프로그램 흐름도  
Fig. 4. Flow chart of pulse compression program using assembly language.

V. 시스템 적용 예

다중 DSP를 이용하여 약 33km의 탐지거리를 갖는 단거리 레이더 및 약 125km의 탐지거리를 갖는 중거리 탐색 레이더의 펄스압축기 구성 가능성을 확인하여 보자. <표 2>는 펄스압축 연산과 관련된 단거리 및 중거리 탐색레이더의 전형적인 설계 파라미터를 나타내고 있다. 펄스압축기 탭 수는 11 및 50으로 하였으며, 거리 셀 수는 각각 540개 및 2,000개로 설정하였다.

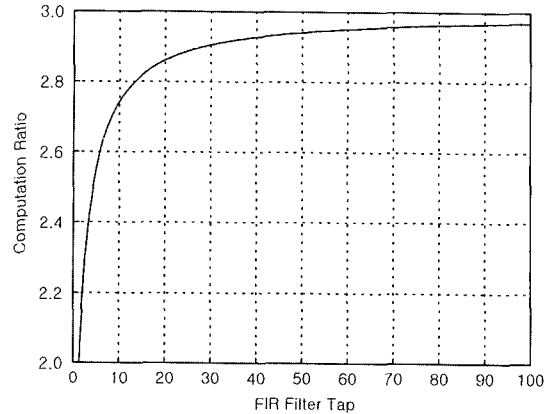


그림 5. C-언어 및 어셈블리어에 대한 펄스압축 연산시간 비

Fig. 5. Computation time ratio of C-language to assembly language for pulse compression.

표 2. 펄스압축기 관련 레이더 파라미터

Table 2. The radar parameters related to pulse compressor.

구분	설계 파라미터	
	단거리 레이더	중거리 레이더
PRF(PRI)	4,500 Hz (222 $\mu$ s)	1,200 Hz (833 $\mu$ s)
송신펄스 폭	4.4 $\mu$ s	20 $\mu$ s
거리 셀	400ns(60m)	400ns(60m)
FIR 필터 탭	11	50
거리 셀 수	540	2,000

<그림 6>은 어셈블리 언어를 사용한 경우, <표 2>의 FIR 필터 탭 수에 대해 수신 거리 셀에 따른 펄스압축 처리의 소요시간으로 단거리 및 중거리 레이더에 대해 각각 676 $\mu$ s 및 10.1ms 가 된다. 따라서 식 (3)에서 분배시간  $T_D$ 가 필요 없는 하드웨어 구조로 설계하는 경우 소요 DSP 수는 각각 3.05 및 12.12가 된다. 따라서 단거리 레이더인 경우에는 4개의 DSP를, 중거리 레이더인 경우에는 13개의 DSP를 사용함으로써 676 $\mu$ s 및 10.1ms 의 시간지연을 갖는 펄스압축기 구성이 가능함을 알 수 있다.

앞에서의 예제는 <그림 2>의 구성과 같이 수신된 신호를 N개의 DSP에 순차적으로 배분함으로써 지연시간은 한 개의 DSP를 사용한 경우의 펄스압축 연산시간이 된다. 따라서 펄스압축 연산에 의한 지연시간을 줄이기 위해서는, <그림 7>과 같이 다수의 DSP를 사용

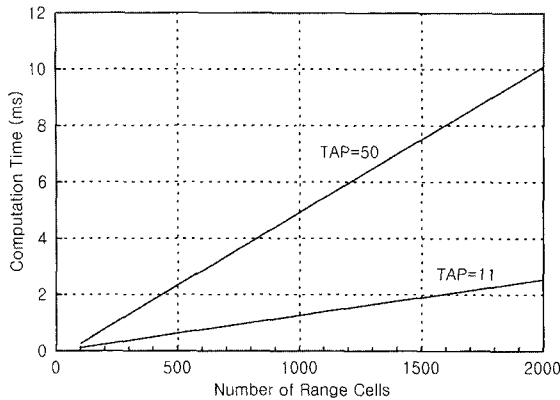


그림 6. 펄스압축기 연산시간  
Fig. 6. Computation time of pulse compressor.

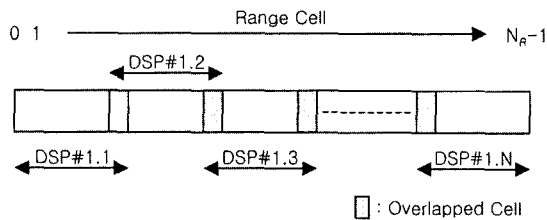


그림 7. 중첩처리를 위한 데이터 분배  
Fig. 7. Data distribution for overlap processing.

하여 거리 셀 데이터를 중첩처리 함으로써 DSP 상호 간 데이터교환 없이 출력신호의 연속성을 유지할 수 있을 것이다.

### VI. 결론

본 논문에서는 다중 DSP를 이용하여 확장성이 용이한 디지털 펄스압축 처리기를 구성하였다. 수신거리 셀 및 FIR 필터 탭 수에 따른 펄스압축 연산시간을 아날로그 디바이스사의 ADSP-21060을 대상으로 C-언어와 어셈블리어언어로 비교·분석하였다. 어셈블리 언어에 대한 분석결과를 탐색 레이더에 적용하여, 펄스압축기 구성에 필요한 DSP 수를 쉽게 예측할 수 있음을 확인하였다. 한편 연산에 따른 지연시간을 줄여야 하는 경우에 적용할 수 있는 거리 셀 데이터 중첩처리 방안을 제시하였다. 본 논문에서 적용한 다중 DSP 펄스압축기 구성은 DSP 한 개를 사용한 경우의 펄스압축 연산시간만 정확히 얻을 수 있으면 소요 DSP 수를 근사적으로 예측할 수 있다.

### 참고 문헌

- [1] D. Curtis Schleher. Electronic Warfare in the Information Age, Artech House, Norwood, MA, pp. 202~213, 1999.
- [2] Stephen To et al., "Digital Implementation Issues in a Pulse Compression Radar System," Information, Decision and Control, IDC-99, Proceedings, pp. 181~186, 1999.
- [3] Ren Peihong, "A Kind of High-Speed Real-Time Digital Pulse Compressor Implemented in Time Domain," Radar, 2001 CIE International Conference on, Proceedings, pp. 966~969, 2001.
- [4] Teng Long and Erke Mao, "High-Speed Real-Time Digital Radar Processor : Design and Implementation," Radar, 2001 CIE International Conference on, Proceedings, pp. 987~991, 2001.
- [5] J. Villasenor and B. Hutchings, "The Flexibility of Configurable Computing," IEEE Signal Processing Magazine, Vol 15, No 5, pp 67~84, 1998.
- [6] J. Tulodziecki, "The Application of Parallel DSP Architectures to Radar Signal Processing," Multiprocessor DSP-Applications, Algorithms and Architectures, IEE Colloquium on, 1995, pp. 5/1~5/7, 1995.
- [7] "ADSP-2106X SHARCTM User's Manual," Analog Devices, Inc., 2nd Edition, 1997.

저 자 소 개



申鉉翼(正會員)

1986년 : 경북대학교 전자공학과 학사. 1988년 : 경북대학교 전자공학과 석사. 1999년~현재 : 충남대학교 대학원 전자공학과 박사과정. 1988년~현재 : 국방과학연구소 선임연구원. <주관심분야 : 레이더 신호처리 및 시험평가 기법개발, 레이더 재밍장치 설계>

호처리 및 시험평가 기법개발, 레이더 재밍장치 설계>



金煥宇(正會員)

1977년 : 서울대학교 전자공학과 학사. 1979년 : 한국과학기술원 전기 및 전자공학과 석사. 1988년 6월 : University of Utah 전자공학과 공학박사. 1980년 6월~현재 : 충남대학교 정보통신공학부 교수. 2000

년~현재 : 대한전자공학회 충남지부 이사. 2002년~현재 : IEEE 대전섹션 부의장. <주관심분야 : 디지털 신호처리, 디지털 통신 >



柳榮鎭(正會員)

1987년 : 경북대학교 전자공학과 학사. 1989년 : 경북대학교 전자공학과 석사. 2002년~현재 : 충남대학교 대학원 전자공학과 박사과정. 1989년~현재 : 국방과학연구소 선임연구원. <주관심분야 : 전자전 신호처리>

호처리>