Journal of KIIT. Vol. 22, No. 10, pp. 79-84, Oct. 31, 2024. pISSN 1598-8619, eISSN 2093-7571 79
http://dx.doi.org/10.14801/jkiit.2024.22.10.79

능동위상배열(AESA) 안테나의 빔 조향을 위한 효율적인 연산기 설계방법

박 인 호*

Efficient Beam Steering Calculator Design of Active Electronically Scanned Array(AESA)

Inho Park*

요 약

AESA(Active Electrically Scanned Array) 안테나의 구성품인 TRB에서는 송신할 때에는 HPA, DRA 수신할 때에는 LNA를 제어하여 RF 신호를 증폭한다. 그리고 MFC(Multi Functional Chip)를 통해 RF 신호의 위상 및 세기를 제어하여 빔 조향을 한다. 그 중 MFC에는 phase shifter 및 attenuator가 있어 원하는 곳에 빔조향을 하기 위해서는 두 부분을 알맞게 제어해야 한다. Attenuator의 입력값은 계산이 아닌 근접전계 측정으로 정해지며, Phase shifter의 입력값은 채널의 위치와 phase gradient 값의 곱을 통해 정해진다. 기존 phase shifter 입력값 계산 방식은 하드웨어 복잡도와 레이턴시가 높은 단점이 있었다. AESA 안테나에서 사용하는 6-bit MFC를 고려하여 phase gradient 연산방법 개선과 그에 따른 설계방안은 하드웨어 복잡도와 레이턴시를 줄이도록 설계하였다.

Abstract

Active Electrically Scanned Array(AESA) Antenna can control beam steering by controlling phase and power control of RF signal through High Power Amplifier(HPA), DRiving Amplifier(DRA), Low Noise Amplifier(LNA) in TRB and Multi Functional Chip(MFC). Phase shifter and attenuator are controlled correctly to steer the beam because MFC has phase shifter and attenuator. Attenuator input is fixed by not calculation but near-field mesurement and phase shifter input is calulcated by multiplying based on channel position and value of phase gradient. Existing calculation method has a feature of complicated digital hardware and higher latency So improved calculation method of phase gradient and designing hardware have feature of reduced complexity and reduced latency by considering of 6-bit MFC used in AESA antenna.

Keywords
AESA antenna, calculation method, digital hardware, efficient design

* 한화시스템 선임연구원
- ORCID: https://orcid.org/0009-0009-9575-5687

· Received: Aug. 30, 2024, Revised: Sep. 23, 2024, Accepted: Sep. 26, 2024
· Corresponding Author: Inho Park
Dept. of Hanwha Systems 491-23, Gyeonggidong-ro, Namsa-eup, Cheoin-gu, Yongin-si, Gyeonggi-do, Repulic of Korea
Tel.: +82-31-8020-7197, Email: ihpark90@hanwha.com

Ⅰ. 서 론

최근 개발되는 항공 레이다 시스템에서는 AESA (Active Electrically Scanned Array) 안테나를 사용하고 있다[1]. 안테나가 차지할 수 있는 공간이 한정적인 항공기에서는 작은 form factor를 가질 수 밖에 없는 환경을 가졌다[2]. 그래서 항공기용 AESA 안테나 TRB에는 내부에 RF 신호를 증폭하여 송신할 수 있는 DRA(DRiving Ampilfier), HPA(High Power Amplifier), 표적에서 반사되어 들어오는 신호를 적은 noise로 증폭시켜주는 LNA(Low Noise Amplifier), 빔 조향을 위해 RF 신호의 phase shifting과 attenuation을 할 수 있는 MFC(Multi Functional Chip) 등의 부품을 가지고 작은 form factor로 제작한다[3][4]. 그림 1에서 보면 송신할 때에는 RF 신호가 HPA, DRA쪽 경로를 통해 송신되며, 수신할 때에는 LNA쪽 경로를 통해 RF 신호가 수신된다. 그리고 MFC가 송수신하는 RF 신호의 위상 및 세기를 조정하는 구조이다. 이러한 부품을 제어하여 신속한 빔조향, 다중 빔 형성 등 다양한 패턴을 형성할 수 있다. 그 중 AESA 안테나에서 가장 중요한 부품은 빔 조향을 가능하게 하는 MFC라고 할 수 있다. 그래서 각 채널의 phase shift 되는 값 및 attenuation 되는 값들은 빔 조향에 있어 가장 중요하다. 그래서 6-bit MFC를 사용할 경우에 입력해야 될 값의 연산은 부품들을 이용하여 물리적으로 안테나를 움직여 빔 조향하지 않고, 안테나 내부의 phase shifter, attenuator를 이용하여 빔 조향을 한다[5][6].

채널 수가 많은 AESA 안테나의 경우 원전계 측정으로 보정은 어렵고 근접전계 측정을 이용하여 채널별 보정을 통해 Attenuator에 입력하는 값을 정하게 된다. 이 값들은 LUT(Look Up Table) 방식으로 안테나 내부에 저장하여 채널별 attenuation 해야 될 값을 지정한다[7]. 하지만 phase shifter에 입력해야할 값은 각 채널별 위치 및 원하는 빔 조향 각도에 따라 계산을 통해 정해지게 된다. 결과적으로 효율적인 phase shifter 연산기를 설계하는 것이 곧 효율적인 안테나 설계를 하는 것이다.

본 논문에서는 6-bit MFC를 사용하였을 때 MFC 입력값을 계산하는 최소 단위에 대한 변환하는 방법으로 제안하였다. II장에서는 AESA 안테나에서 빔 조향을 위해 사용하는 계산식을 설명하고, 6bit MFC를 사용하였을 때 계산식을 최적화하여 하드웨어 리소스를 줄일 수 있는 방안에 대해 설명한다. III장에서는 효율적인 빔 조향을 위해 연산기 설계 방법에 대해서 설명하고, IV장에서는 기존 연산기 구조 및 리소스 사용량을 비교하여 리소스를 얼마나 덜 사용하였는지 확인한다.

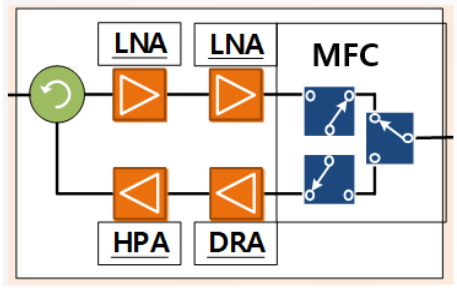

그림 1. TRB (Transmit-Receiver Block) 구조
Fig 1. TRB (Transmit-Receiver Block) structure

Ⅱ. AESA 안테나 빔조향 계산식

우선, AESA 안테나에서 phase shifter에 입력하는 값들을 계산하기 위해서 구 좌표계 대신에 u,v좌표계를 사용한다[8]. u,v 좌표계는 그림 2를 통해 볼 수 있으며, 3차원 공간에 있는 표적을 x,y plane에 매핑하기 위해 방위각(Azimuth)와 고각(Elevation)을 사용한다.

$$u = \sin\theta \times cos\phi \qquad (1)$$
$$v = \sin\phi$$

식 (1)에서의 θ는 Azimuth (deg) 값이며, ϕ는 Elevation (deg) 값이다. 기존에는 u,v 좌표계를 사용하여 식 (2)의 계산식으로 phase gradient를 계산한다.

$$PG_U = 360 \times \lambda \times dh \times u \qquad (2)$$
$$PG_V = 360 \times \lambda \times dv \times v$$
$$(\lambda = f/c)$$

식 (2)에서의 f는 송수신하는 RF 신호의 주파수, c는 빛의 속도를 의미한다. 식 (3)과 같이 계산된 phase gradient를 바탕으로 안테나에서의 채널별 위

치 값(x_{pos}, y_{pos})을 곱하여 phase shifter에 입력할 값을 결정하게 된다. 그림 3에서 파란색 숫자는 TRM의 번호를 의미하며, 그 아래 하늘색 배경의 검은색 숫자는 채널번호를 의미한다. 채널위치 값은 채널별 X축 간격과 Y축 간격과 상대 위치값을 곱하여 생성되게 된다. 예를 들어 그림 3을 보면 TRM 0번의 12번 채널의 X축 위치는 (X축 채널 간격) * 31이며, Y축 위치는 (Y축 채널 간격) * 8로 정해진다.

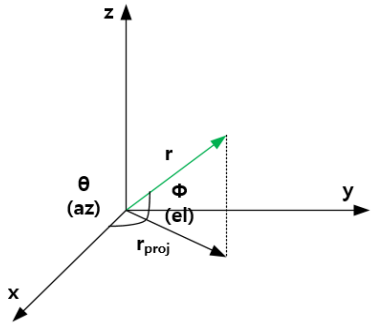

그림 2. u,v 좌표계
Fig. 2. u,v coordinate

$$output_{\deg} = PG_U \times x_{pos} + PG_V \times y_{pos} \tag{3}$$

하지만, phase shifter의 입력해야 하는 값은 360도마다 반복 하며, 입력 bit width에 따라서 제어할 수 있는 각도의 최소값 한계가 있으므로 6bit phase shifter에 입력해야 할 값은 아래 식 (4)와 같이 정해진다.

$$\begin{aligned} output_{bit} &= (output_{\deg} \% 360) / (360/2^6) \\ &= (output_{\deg} \% 360) / 5.625 \end{aligned} \tag{4}$$

식 (4)를 보다시피 360의 나머지와 360을 2^6로 나누는 연산은 2진수로만 이뤄진 디지털 하드웨어 시스템에서 구현하기 어렵게 되어있다. 앞서 기술한 방법은 기존 방법이며, 설계한 시스템에서는 6-bit MFC의 최소 단위는 5.625도 이기 때문에 PG_U, PG_V를 계산하는 식의 최소 단위를 1도가 아닌 5.625도로 변경하여 식 (5)와 같이 개선하였다. 식 (5)에서 볼 수 있다시피 64를 곱하는 것은 bit shift만으로 수행할 수 있기 때문에 디지털 하드웨어에서 리소스 및 레이턴시(latency)를 줄일 수 있게 된다. 식 (5)의 연산으로 각 채널별로 계산한 값들의 예시는 그림 4에서 볼 수 있다.

$$\begin{aligned} PG_U &= 360 \times \lambda \times dh \times u \div 5.625 \\ &= 64 \times \lambda \times dh \times u \\ PG_V &= 360 \times \lambda \times dv \times v \div 5.625 \\ &= 64 \times \lambda \times dv \times v \end{aligned} \tag{5}$$

일반적인 레이다 시스템의 경우 PG_U, PG_V 값을 다른 장비로 받아 동작하는 경우가 많다. 이 경우에는 해당 값들에 할당할 수 있는 bit width가 정해져 있게 되며, 그러한 시스템에서는 기존에 적용하였던 계산 방법보다 scale 할 수 있는 값의 범위가 좀 더 커진다.

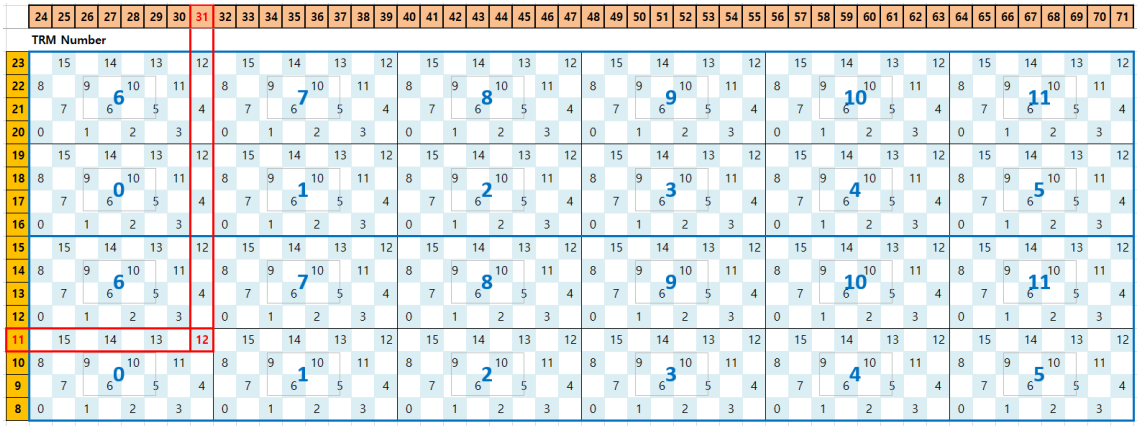

그림 3. 채널별 위치값
Fig. 3. Channel position index

PG_U	16728	scaled bit									Difference value MAX	1
PG_V	41747	scaled bit									Difference value SUM	1

Array #	TRPA ID	TRM ID		Index #		⊿x *x + ⊿y *y	ROUND(/scaling)			Least Significant 6bit	Optimal value	Difference value with optimal
		TRB ID	Sub Array #	x	y		>> 10	Round down	MOD			
1		0	0	47	7	1078445	1053	526	1	15	15	0
2			1	45	7	1044989	1020	510	0	62	62	0
3			2	43	7	1011533	987	493	1	46	46	0
4			3	41	7	978077	955	477	1	30	30	0
5			4	40	6	919602	898	449	0	1	1	0
6			5	42	6	953058	930	465	0	17	17	0
7			6	44	6	986514	963	481	1	34	34	0
8			7	46	6	1019970	996	498	0	50	50	0
9			8	47	5	994951	971	485	1	38	38	0
10			9	45	5	961495	938	469	0	21	21	0
11			10	43	5	928039	906	453	0	5	5	0
12			11	41	5	894583	873	436	1	53	53	0
13			12	40	4	836108	816	408	0	24	24	0
14			13	42	4	869564	849	424	1	41	41	0
15			14	44	4	903020	881	440	1	57	57	0
16			15	46	4	936476	914	457	0	9	9	0

그림 4. Floating point 연산을 적용했을 때와 Fixed point 연산을 적용하였을 때 6bit MFC 입력값 비교
Fig. 4. Comparison of 6-bit MFC input values with floating point versus fixed point operations

그렇게 되면 PG_U, PG_V 값을 가지고 phase shifter에 입력해야 될 값을 연산할 때 좀 더 정확하게 연산할 수 있게 된다. 그리고 64와 같이 2의 제곱수가 곱하는 식으로 돼 bit shift 하는 것만으로 간단해져 따로 곱셈기를 사용하지 않아 디지털 하드웨어 리소스를 적게 사용하며 레이턴시를 줄일 수 있는 결과를 만들 수 있다.

III. 연산법별 Phase shift 입력값 비교

각 채널별 현재 구성되어 있는 레이다 시스템에서는 PG_U, PG_V의 bit width는 16bit로 제한되어 있어 계산 방법별로 scaling 할 수 있는 상한값이 정해져 있다. 그래서 위와 같은 상황에서 fixed point 연산하는 경우와 이상적으로 연산하는 방법인 floating point로 연산한 결과와 비교해보았다. floating point로 연산하는 방법은 하드웨어의 사용량과 관계없이 구현된 결과이기 때문에 MFC에 좀 더 정확한 Phase shift 해야할 값을 구해낼 수 있다. 기존 방법으로 연산한 값과 제안하는 방법의 MFC 입력값 오차들을 표 1, 2에 적어놓았다. 입력값의 오차는 floating point로 연산한 MFC 입력값과 기존방법 및 제안하는 방법의 차이를 말하는 것이다. 확인해 본 빔 조향각도에서 MFC에 입력값의 전체채널의 오차합이 제안하는 방법에서 줄어듦을 확인할 수 있었다. 이는 scaling 할 수 있는 범위가 넓어지기 때문이라고 볼 수 있을 것이다.

표 1. 계산방법별 6-bit Phase shifter 입력 오차 분석 (방위각 = 45º, 고각 = 45º)
Table 1. Phase shifter input error comparison (Azimuth = 45º, Elevation = 45º)

6-bit Phase shifter	Legacy method	Proposed method
Maximum error	1	1
Error in total	17	1

표 2. 계산방법별 6-bit Phase shifter 입력 오차 분석 (방위각 = 60º, 고각 = 10º)
Table 1. Phase shifter input error comparison (Azimuth = 60º, Elevation = 10º)

6-bit Phase shifter	Legacy method	Proposed method
Maximum error	1	1
Error in total	27	14

IV. 하드웨어 설계 및 검증

기존의 설계 방법과 제안하는 설계 방법의 리소스 사용량 및 레이턴시를 비교하여 얼마나 효율적으로 설계되었는지 확인한다. 해당 검증은 현재 시스템에서 사용하고 있는 FPGA인 xilinx의 xc7a200tsbg 484-1 FPGA 사용환경에서 검증하였다. 그리고 해당 implementation 결과는 Vivado 2020.2 버전에서 확인하였다.

연산기만의 하드웨어 리소스의 사용량을 비교해보았을 때에는 표 3과 같이 사용량이 92%((960-82)/960) 감소하였음을 볼 수 있다.

2진수 연산에 최적화 된 연산방식을 적용하였기 때문이라고 볼 수 있다. 현재 구현방식에서는 해당 연산기는 채널마다 구현되어있다. 그래서 실제 시스템에서 16채널을 적용하고 있기 때문에 연산기 부분만 봤을 때에는 LUT 사용량 차이가 16배 나며 크게 리소스를 줄였다고 볼 수 있다.

표 3. 계산방법별 하드웨어 리소스 사용량 비교
Table 3. Calculator resource usage comparison between methods

Resource usage	Legacy method	Proposed method
LUT usage	960	82
Resource usage difference (Proposed / Legacy)	8.54%	

또한 전체 하드웨어에서의 리소스도 함께 감소하였으며, 이는 그림 5, 6과 표 4를 통해 볼 수 있다. 그림 5, 6에서 하늘색으로 보이는 부분이 FPGA에서 LUT가 차지하는 부분이다. 제안하는 연산기 구조를 사용하였을 때 기존 방법을 적용하였을 때보다 비해 하드웨어 리소스가 37.5% 감소하였음을 확인하였다.

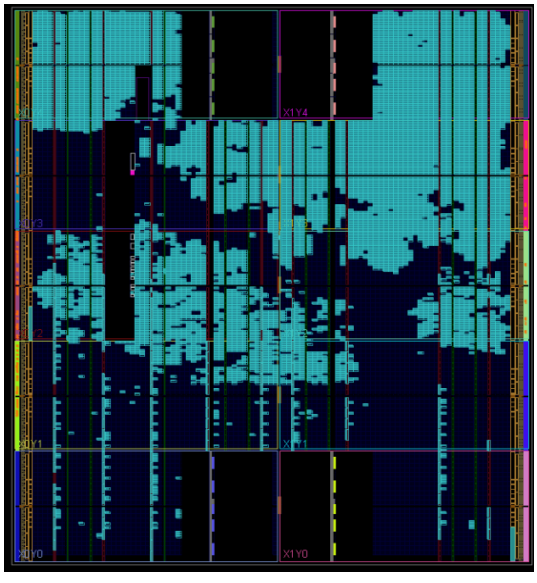

그림 5. 기존 방법으로 구현한 FPGA implementation 결과
Fig. 5. FPGA implementation result using legacy method

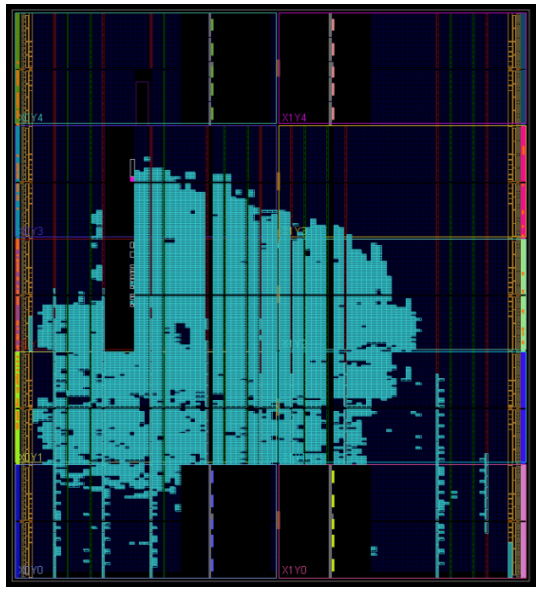

그림 6. 수정된 방법으로 구현한 FPGA Implementation 결과
Fig. 6. FPGA implementation result using proposed method

표 4. 전체 하드웨어에서의 리소스 사용량 비교
Table 4. Entire resource usage comparison between methods

Resource usage	Legacy method	Proposed method
LUT usage	39371	24605
Resource usage difference (Proposed / Legacy)	62.5%	

레이턴시 또한 기존에 비해 표 5에서 보듯이 절반 가까이 줄어들었음을 확인할 수 있다. 레이턴시가 줄게 되면 연산한 결과가 나오는 데에 적은 시간이 걸리게 되어 좀 더 빠르게 빔 조향이 되는 결과를 가져올 수 있게 된다.

표 5. 계산방법별 연산기 레이턴시 비교
Table 5. Calculator latency comparison between methods

Resource usage	Legacy method	Proposed method
latency(ns)	17.935	8.799
latency difference (Proposed / Legacy)	49%	

V. 결 론

하드웨어 리소스를 줄이면서 빔 조향 성능에 영향을 미치지 않도록 설계해보았다. 이미 설계되어 있는 하드웨어 구조상 MFC에 phase shifter에 입력되는 bit width는 정해져 있음을 고려하여 연산방식을 개선한 방법이었다. 성능에도 영향을 미치지 않으면서 리소스 및 레이턴시를 줄이는 결과를 확인하였다. 추후 bit width가 변경된다면 고려하여 연산기를 설계해야 할 것이다. 해당 방법은 레이턴시를 절반가까이 줄이면서, 리소스 또한 기존방법에 비해 전체 하드웨어의 리소스를 37.5% 감소할 수 있음을 보였다.

그리고 추후 개선할 수 있는 방안으로는 각 채널별 곱셈기를 따로 두었지만, 곱셈기는 하드웨어 리소스를 많이 차지하는 연산기이다. 이를 대신하여 그림 4에서 보다시피 채널별 위치값은 상대적이므로 기준 위치값을 입력되면 다른 채널의 위치값들을 계산하는 것은 더하는 방식으로 대체할 수 있을 것이다. 이러한 방법들은 지금의 구조보다도 더 리소스를 감소시키면서 레이턴시를 줄일 수 있는 방법이다. 추후에 해당 방법을 적용하여 설계를 해보는 것도 고려를 해야할 것이다.

References

[1] W. Gruener, J. P. Toernig, and P. J. Fielding, "Active-electronically-scanned-array based radarsystem feature", Radar Systems(Radar 97), No. 449, pp. 339-343, 1997. https://doi.org/10.1049/cp:19971691.

[2] A. M. Kinghorn, "Where Next For Airborne AESA Technology?", IEEE Aerospace and Electronic Systems Magazine, Vol. 24, No. 11, pp. 16-21, Nov. 2009. https://doi.org/10.1109/MAES.2009.5344177.

[3] S. Choi, B. Lee, and Y. Choi, "Transmit/Receive Module Amplitude/Phase Correction Algorithm Design for Improving Phased Array Antenna Beam Performance", Journal of KIIT, Vol. 20, No. 5, pp. 71-79, May 2022. http://dx.doi.org/10.14801/jkiit.2022.20.5.71.

[4] H. Jung, J. Han, M. Kang, C. Lee, B. Park, Y. Kim, and H. Kim, "Design and Fabrication of X-Band Air-Cooled, Compact, and Lightweight, Active Electronically Scanned Array Antenna Unit for Aircraft Installation", The Journal of KIEES, Vol. 34, No. 8, pp. 618-628, Aug. 2023. https://doi.org/10.5515/KJKIEES.2023.34.8.618.

[5] R. Sundararaman, "Implementation of optimized 6-bit phase angle calculation from phase gradients for T/R Modules in active phased array radars using FPGA", 2009 International Conference on Control, Automation, Communication and Energy Conservation, Perundurai, India, Jun. 2009.

[6] U. Nickel, "Fundamentals of Singal Processing for Phased Array Radar", NATO Science and Technology Organization Inc., pp. 1-22, Sep. 2006

[7] D. Jang, "The Calibration Method of Airborne Active Electrically Scanned Array(AESA) Antenna using Near-field Measurement", Journal of KIIT, Vol. 21, No. 12, pp. 121-128, Dec. 2023. https://doi.org/10.14801/jkiit.2023.21.12.121.

[8] J. Wang, J. Li, B. Sun, and Z. Zuo, "SAR image synthesis based on conditional generative adversarial networks", The Journal of Engineering, IET, Vol. 2019, No. 21, pp. 8093-8097, Nov. 2019. https://doi.org/10.1049/joe.2019.0696.

저자소개

박 인 호 (Inho Park)

2015년 2월 : 충남대학교 전자공학과(공학사)
2017년 8월 : 충남대학교 전자공학과(공학석사)
2017년 9월 ~ 현재 : 한화시스템 연구원
관심분야 : 디지털 하드웨어, FPGA, 레이다 Hardware