

# 시너지 설계: 소프트웨어 정의 라디오 시스템의 생애주기 가치 극대화를 위한 DFX 방법론 적용

## 서론: 유연성을 넘어 - 소프트웨어 정의 세계의 물리적 기반

소프트웨어 정의 라디오(SDR)의 본질적 특징은 소프트웨어를 통해 기존 하드웨어가 수행하던 무선 통신 기능을 구현함으로써 얻어지는 비할 데 없는 유연성에 있습니다.<sup>1</sup> 이러한 유연성은 단일 하드웨어 플랫폼이 다양한 통신 표준, 주파수 대역, 파형을 동적으로 지원할 수 있게 하여, 기술 발전과 변화하는 요구사항에 신속하게 대응할 수 있는 능력을 부여합니다.<sup>3</sup> 그러나 이 소프트웨어적 유연성은 결코 추상적인 개념이 아니며, 그 잠재력은 전적으로 견고하고 효율적인 하드웨어 플랫폼에 의존합니다. SDR의 진정한 가치, 신뢰성, 그리고 총소유비용(Total Cost of Ownership, TCO)은 소프트웨어의 능력뿐만 아니라, 그 물리적 설계와 아키텍처에 적용된 선견지명에 의해 결정됩니다.

이 지점에서 DFX(Design for X, 목적 지향 설계)는 SDR 개발의 핵심적인 공학 철학으로 부상합니다. DFX는 제품의 전체 생애주기에 걸쳐 특정 목표(X)를 최적화하기 위한 체계적인 설계 접근법을 의미하며, 여기서 'X'는 제조(Manufacturing), 테스트(Testability), 신뢰성(Reliability), 서비스 용이성(Serviceability) 등 다양한 속성을 포함합니다.<sup>4</sup> DFX는 하드웨어가 SDR의 잠재력을 제약하는 요소가 아니라, 오히려 그 잠재력을 전체 생애주기 동안 안정적이고 효율적으로 발현시키는 강력한 조력자가 되도록 보장하는 규율입니다. 이는 영리한 프로토타입을 대량 생산 가능하고, 테스트할 수 있으며, 신뢰할 수 있고, 서비스 가능한 상용 제품으로 전환시키는 핵심 과정입니다.

특히 SDR의 맥락에서 DFX는 전통적인 하드웨어 설계의 경계를 넘어섭니다. 서비스 용이성이나 테스트 용이성과 같은 원칙들은 물리적인 하드웨어 기능과 정교한 소프트웨어 아키텍처의 긴밀한 통합을 통해 실현됩니다. 예를 들어, 원격 펌웨어 업데이트(OTA) 기능은 소프트웨어적으로 관리되지만, 이를 안정적으로 지원하기 위해서는 전원 차단 시에도 데이터 무결성을 보장하는 메모리 컨트롤러와 복구 가능한 부트로더 같은 하드웨어적 기반이 필수적입니다. 본 보고서는 SDR 기술과 DFX 철학의 이러한 공생 관계를 심층적으로 탐구하고, DFX 원칙을 SDR 시스템에 적용했을 때 얻을 수 있는 구체적인 기대 효과를 분석하며, 이를 실제 엔지니어링 현장에 적용할 수 있는 실용적인 가이드를 제공하는 것을 목표로 합니다.

## 섹션 1: SDR 시스템 설계에서 DFX의 전략적 필요성

SDR 개발 프로젝트, 특히 군사, 항공우주, 통신 인프라와 같이 미션 크리티컬하거나 상업적으로 중요한 분야에서 DFX는 단순히 '있으면 좋은 것'이 아니라, 프로젝트의 성패를 좌우하는 전략적 필수 요소입니다. 이 섹션에서는 왜 DFX가 SDR 개발의 초기 단계부터 반드시 고려되어야 하는지를 전략적 관점에서 논증합니다.

### 1.1. "소프트웨어 정의"의 전체 생애주기 비용

SDR 시스템의 초기 개발 비용, 즉 비반복적 엔지니어링(Non-Recurring Engineering, NRE) 비용은 전체 생애주기 비용에 비하면 빙산의 일각에 불과한 경우가 많습니다. 제품의 전체 생애주기 비용에는 제조, 테스트, 배포, 유지보수, 업그레이드, 기술 지원 등 막대한 비용이 포함됩니다. DFX는 설계 초기 단계에 이러한 후속 단계의 요구사항을 미리 반영함으로써 총소유비용을 근본적으로 절감하는 것을 목표로 합니다.<sup>5</sup>

일반적으로 제품 생애주기 비용의 약 70%에서 95%가 설계 단계에서 결정된다는 원칙은 SDR 시스템에 더욱 극적으로 적용됩니다.<sup>5</sup> 예를 들어, 초기 설계에서 미래의 고성능 파형 처리를 감당하지 못하는 FPGA를 선택하거나, 발열 문제를 고려하지 않은 열악한 열 설계를 채택하는 경우를 생각해 볼 수 있습니다. 이러한 결정은 당장의 부품 비용을 절감할 수는 있겠지만, 미래에 새로운 소프트웨어 기능을 추가하려 할 때 성능 저하를 유발하거나 심지어는 전체 하드웨어 플랫폼을 재설계해야 하는 막대한 비용을 초래할 수 있습니다. DFX는 이러한 근시안적 최적화를 방지하고, 장기적인 관점에서 시스템의 가치를 극대화하는 전략적 의사결정을 가능하게 합니다.

### 1.2. DFX가 해결하는 SDR 개발의 고유한 과제

SDR 개발은 전통적인 전자제품 설계와는 다른 고유한 기술적 과제들을 안고 있으며, DFX 원칙들은 이러한 문제들을 해결하는 데 특화된 솔루션을 제공합니다.

- 과제 1: 고밀도, 혼합 신호의 복잡성

SDR 플랫폼은 고주파 RF 아날로그 회로, 고속 디지털 로직, 그리고 정밀한 전력 관리 회로를 단일 PCB 위에 고도로 집적합니다.<sup>6</sup> 이러한 혼합 신호 환경은 아날로그 단과 디지털 단 간의 노이즈 커플링, 임피던스 매칭, 신호 무결성 문제 등을 야기하며, 이는

제조(DFM), 테스트(DFT), 신뢰성(DFR) 측면에서 심각한 도전 과제가 됩니다. 예를 들어, 고성능 ADC/DAC 주변의 전력 노이즈는 시스템의 동적 범위(Dynamic Range)를 저하시키고, FPGA의 고속 인터페이스 배선 오류는 데이터 손실을 유발할 수 있습니다. DFX는 이러한 문제들을 초기 설계 단계에서 체계적으로 검토하고 완화하기 위한 가이드라인과 분석 도구를 제공합니다.<sup>8</sup>

- **과제 2: 장기 현장 운용 및 표준의 진화**  
군용 통신, 위성, 그리고 5G/6G와 같은 통신 인프라에 사용되는 SDR은 수십 년에 달하는 긴 운용 수명을 가집니다.<sup>3</sup> 이 기간 동안 새로운 통신 표준이 등장하고, 보안 위협이 진화하며, 요구되는 기능이 변화하는 것은 필연적입니다. 따라서 이러한 **시스템은 단순히 고정된 기능을 수행하는 장비가 아니라, 지속적으로 진화하고 적응해야 하는 플랫폼입니다.** 이는 **서비스 용이성 설계(Design for Serviceability, DFS)와 모듈화 설계(Design for Modularity)를 매우 중요하게 만듭니다.** 현장에서의 소프트웨어 업데이트는 물론, 필요에 따라 RF 모듈과 같은 하드웨어 구성 요소를 교체하거나 업그레이드하는 시나리오까지 고려한 설계가 요구됩니다.
- **과제 3: 성능-전력-발열의 삼각 딜레마**  
최신 SDR이 요구하는 복잡한 파형과 높은 데이터 처리량을 실시간으로 구현하기 위해서는 고성능 FPGA, SoC, DSP가 필수적입니다.<sup>11</sup> 이러한 고성능 프로세서들은 상당한 양의 전력을 소비하며, 이는 곧 막대한 열 발생으로 이어집니다.<sup>13</sup> 적절한 열 관리가 이루어지지 않으면, 칩의 온도가 상승하여 최대 성능을 발휘하지 못하는 '성능 스로틀링(Performance Throttling)' 현상이 발생하거나, 장기적으로는 부품의 수명이 단축되어 시스템 고장의 원인이 됩니다. 신뢰성 설계(Design for Reliability, DFR), 특히 열 관리 및 전력 공급 설계는 SDR이 명시된 성능을 안정적으로, 그리고 장기간 유지하기 위한 핵심적인 전제 조건입니다.

이러한 전략적 중요성을 한눈에 파악할 수 있도록, 다음 표는 주요 DFX 원칙들이 SDR 시스템의 각 구성 요소에 어떻게 적용되는지를 요약하여 보여줍니다. 이는 이어지는 상세 분석의 로드맵 역할을 합니다.

표 1: SDR 시스템 구성 요소에 매핑된 DFX 원칙

DFX 원칙	RF 프론트엔드 (LNA, PA, 필터 등)	ADC/DAC	디지털 백엔드 (FPGA/CPU)	전력 시스템	기구/인클로저	소프트웨어/펌웨어 스택
<b>DFM/DFA</b> (제조/조립)	표준 RF 커넥터(SMA 등) 사용, 솔드 캔의 자동화된	미세 피치 패키지의 솔더링 공정 최적화, 주변 부품	BGA 패키지 솔더링 검사를 위한 AOI/AXI	고전력 부품의 열 방출을 고려한 PCB 레이아웃	조립 용이성을 위한 모듈식 구조, 표준화된	제조 단계 테스트 펌웨어의 용이한 프로비저닝

	실장 고려	배치 간격 확보	접근성 확보		패스너 사용	
<b>DFT (테스트)</b>	주요 RF 경로에 테스트 포인트 또는 커플러 배치, RF 루프백 경로 구현	아날로그 성능 측정을 위한 테스트 포인트, 디지털 인터페이스 테스트용 JTAG	JTAG 바운더리 스캔, FPGA 내장 로직 분석기(BIST), 디버그 포트	모든 전압 레일에 대한 테스트 포인트, 전류 모니터링 회로	테스트 픽스처와의 용이한 결합을 위한 설계	원격 진단을 위한 상태 모니터링 API, 내장 자체 테스트 루틴
<b>DFR (신뢰성)</b>	PA의 열 관리, ESD 보호 회로, 고신뢰성 RF 부품 사용	전원 노이즈 필터링 강화, 안정적인 클럭 소스 공급	FPGA/CPU의 열 관리(히트싱크, 팬), 복구 가능한 부트로더 (골든 이미지)	전원부 서지 보호, 고품질 커패시터 사용, 전력 효율 최적화	진동 및 충격에 대한 내구성 확보, 효과적인 열 방출 구조	위치독 타이머 구현, 메모리 오류 검출 및 수정 코드
<b>DFS (서비스)</b>	주파수 대역 변경을 위한 모듈형 RF 도터보드 설계	-	OTA 펌웨어/게이트웨어 업데이트 지원, JTAG을 통한 현장 복구	교체 용이성을 위한 모듈형 전원 공급 장치	주요 부품에 대한 접근 용이성, 분해/재조립 용이성	모듈식 소프트웨어 아키텍처, 원격 구성 및 관리 기능

## 섹션 2: 핵심 DFX 원칙과 SDR에 대한 기대 효과

이 섹션에서는 각 핵심 DFX 원칙을 심층적으로 분석하고, 이러한 원칙들이 SDR 시스템에

적용되었을 때 구체적으로 어떤 실질적인 효과를 가져오는지 상세히 설명합니다.

## 2.1. 제조 및 조립 용이성 설계 (DFM/DFA): 물리적 플랫폼 최적화

DFM(Design for Manufacturing)은 개별 부품들을 쉽고 비용 효율적으로 생산할 수 있도록 최적화하는 것이고, DFA(Design for Assembly)는 전체 제품을 쉽게 조립할 수 있도록 최적화하는 것입니다.<sup>14</sup> 이 두 원칙의 궁극적인 목표는 제조 비용 절감, 시장 출시 기간 단축, 그리고 품질 향상입니다.<sup>15</sup>

### SDR 특화 DFM 적용 방안

- 부품 선정: SDR 설계 시, 공급망 위험을 줄이고 규모의 경제를 활용하기 위해 표준화되고 쉽게 구할 수 있는 수동 소자(저항, 커패시터), 커넥터 등을 우선적으로 고려해야 합니다.<sup>14</sup> 특히 RF 분야에서는 널리 사용되는 SMA 커넥터를 U.FL과 같은 특수 커넥터보다 우선적으로 고려하거나, 다양한 제조사에서 공급하는 표준 풋프린트의 FPGA 또는 SoC를 선택하는 것이 장기적으로 유리합니다. 이는 특정 부품의 단종이나 공급 부족 사태로부터 프로젝트를 보호하는 중요한 전략입니다.
- 제조 공정을 고려한 PCB 레이아웃: SDR의 복잡한 PCB는 표준 제조 공정과 호환되도록 설계되어야 합니다. 이는 여러 개의 보드를 하나의 큰 패널에 배열하여 생산 효율을 높이는 패널라이제이션, 자동화 장비가 보드의 위치를 정확히 인식하기 위한 기준점인 피두셜 마크의 배치, FPGA와 같은 미세 피치 부품의 솔더링 불량을 방지하기 위한 솔더 마스크 설계, 그리고 자동화된 조립 및 검사 장비의 접근을 위한 부품 간 충분한 이격 거리 확보 등을 포함합니다.<sup>15</sup> 특히 아날로그 RF 회로와 고속 디지털 회로가 공존하는 SDR의 혼합 신호 레이아웃에서는 두 영역 간의 접지 분리, 전원 분리, 신호 격리 등을 통해 상호 간섭을 최소화하는 설계가 필수적입니다.

### SDR 특화 DFA 적용 방안

- 모듈식 설계: Ettus Research의 USRP 제품군에서 흔히 볼 수 있는 RF 프론트엔드용 도터보드(daughterboard) 구조는 DFA의 훌륭한 예시입니다.<sup>19</sup> 이 접근법은 메인 디지털 보드를 재설계하지 않고도 RF 도터보드 교체만으로 주파수 대역을 변경하거나 성능을 업그레이드할 수 있게 해줍니다. 이는 조립 공정을 단순화하고, 제품 라인업의 유연성을 극대화하며, 수리 및 유지보수를 용이하게 만듭니다.
- 커넥터 및 케이블링 전략: 조립 과정에서 오류 발생 가능성이 높고 시간이 많이 소요되는

수작업 납땜 케이블 연결을 최소화해야 합니다. 대신, 견고하고 조립이 간편한  
보드-투-보드 커넥터나 표준화된 플렉서블 케이블(Flex Cable)을 사용하는 것이  
바람직합니다. 또한, 커넥터의 방향이 명확하여 오조립을 방지할 수 있는 설계를 채택하는  
것이 중요합니다.

## 기대 효과

DFM/DFA 원칙을 충실히 적용함으로써 SDR 시스템은 단위당 생산 비용 절감, 제조 수율 향상,  
조립 과정에서의 오류 감소, 그리고 예측 가능하고 탄력적인 부품 공급망 확보라는 실질적인  
이점을 얻을 수 있습니다. 이는 곧 더 높은 이윤과 더 빠른 시장 진입으로 이어집니다.

## 2.2. 테스트 용이성 설계 (DFT): 실리콘부터 시스템까지의 검증 가능성 확보

DFT(Design for Testability)는 테스트를 더 쉽고, 저렴하며, 더 철저하게 수행할 수 있도록 설계  
단계에서 의도적으로 특정 기능을 추가하는 활동입니다.<sup>20</sup> DFT의 핵심은 회로 내부 노드에 대한  
제어 가능성(Controllability)과 관찰 가능성(Observability)을 높이는 것입니다.

### SDR 특화 DFT 적용 방안 (다계층 접근법)

SDR의 복잡성을 고려할 때, DFT는 단일 계층이 아닌 여러 계층에 걸쳐 체계적으로 적용되어야  
합니다.

- 부품 레벨 DFT: FPGA, 프로세서, CPLD와 같은 복잡한 집적회로(IC)에 내장된 JTAG(Joint Test Action Group / IEEE 1149.1) 인터페이스의 활용이 핵심입니다. JTAG의 바운더리 스캔(Boundary Scan) 기능을 사용하면, 물리적인 프로브 접촉 없이도 IC 핀 주변의 납땜 불량(단선, 단락)을 검출할 수 있습니다. 이는 고밀도 BGA 패키지가 사용된 보드의 생산 테스트에 필수적입니다. 또한, JTAG은 IC 내부 로직을 디버깅하고 펌웨어를 프로그래밍하는 통로로도 사용되어 개발 및 유지보수 과정에서 매우 중요한 역할을 합니다.<sup>22</sup>
- 보드 레벨 DFT: 주요 전압 레일, RF 신호 경로, 중요한 제어 신호 등 아날로그 및 디지털 신호에 대한 물리적인 테스트 포인트를 전략적으로 배치해야 합니다. 이를 통해 개발 및 생산 과정에서 오실로스코프나 로직 분석기 같은 계측 장비를 쉽게 연결하여 신호를 측정하고 문제를 진단할 수 있습니다. 또한, 보드 전체를 자동으로 테스트하는 "베드 오브 네일즈(bed-of-nails)" 또는 플라잉 프로브(flying probe) 테스터와 호환되도록 설계하는

것이 중요합니다.

- 시스템 레벨 **DFT**: 이 영역에서 **SDR**의 소프트웨어 정의 특성이 빛을 발합니다.
  - 하드웨어 기반 소프트웨어 테스트: 송신(TX) 출력단에서 감쇠기(attenuator)를 거쳐 수신(RX) 입력단으로 신호를 되돌리는 RF 루프백(loopback) 경로를 하드웨어적으로 구현하고, 이를 소프트웨어 명령으로 활성화할 수 있도록 설계합니다. 이를 통해 장비가 현장에 설치된 상태에서도 전체 송수신 체인의 기능적 무결성을 원격으로 검증하는 포괄적인 자체 테스트를 수행할 수 있습니다.
  - 소프트웨어/API 레벨 테스트: 온도, 전압, 팬 속도와 같은 하드웨어 센서의 측정값을 소프트웨어 API를 통해 외부에서 조회할 수 있도록 제공합니다. USRP 하드웨어 드라이버(UHD)의 센서 API가 좋은 예시로, 이를 통해 원격으로 시스템의 건강 상태를 모니터링하고 잠재적인 문제를 사전에 진단할 수 있습니다.<sup>24</sup>
  - 내장 자체 테스트 (**BIST, Built-in Self-Test**): **FPGA 내부에 특정 로직(예: 메모리 블록, 고속 직렬 인터페이스)을 자체적으로 테스트하는 회로를 구현**합니다. 시스템 부팅 시 BIST가 자동으로 실행되고, 그 결과를 호스트 프로세서에 보고하여 하드웨어의 초기 상태를 신속하게 확인할 수 있습니다.<sup>21</sup>

## 기대 효과

다계층 DFT 전략을 적용하면 생산 단계에서의 테스트 시간과 비용을 획기적으로 절감할 수 있습니다. 또한, 결함 탐지율을 높여 불량 제품의 출고를 방지하고, 원격 진단 및 현장 문제 해결 능력을 향상시켜 전체적인 유지보수 비용을 줄일 수 있습니다. 이는 제품의 품질과 신뢰성에 대한 고객의 신뢰를 높이는 직접적인 요인이 됩니다.

## 2.3. 신뢰성 설계 (DFR): 견고하고 복원력 있는 SDR 아키텍처 구축

DFR(Design for Reliability)은 제품의 전체 수명 동안 고장을 예방하고 안정적인 작동을 보장하기 위해 설계 단계부터 신뢰성을 체계적으로 구축하는 사전 예방적 프로세스입니다.<sup>5</sup>

### SDR 특화 DFR 적용 방안

- 열 관리: 고성능 SDR에서 가장 중요한 DFR 요소 중 하나는 열 관리입니다.
  - 문제 분석: 최신 SoC와 FPGA는 복잡한 연산을 수행하며 상당한 열을 발생시킵니다. 부적절한 열 설계는 칩의 온도를 위험 수준까지 상승시켜 성능 저하를 일으키고, 장기적으로는 열 스트레스로 인해 부품의 수명을 급격히 단축시키는 주요 원인이

됩니다.<sup>13</sup>

- 구현 방안: **FPGA**, 전력 증폭기(PA)와 같은 주요 발열 부품에 적절한 크기와 형태의 히트싱크를 부착하고, 시스템 내부에 원활한 공기 흐름을 만들어 열을 효과적으로 배출할 수 있도록 팬의 위치와 인클로저의 통풍구를 설계해야 합니다. 또한, PCB 설계 시 열 전도성이 높은 서멀 비아(thermal via)를 사용하여 칩에서 발생하는 열을 PCB의 넓은 접지면으로 분산시키는 기법도 매우 효과적입니다.
- 전력 시스템 설계: **안정적이고 깨끗한 전원 공급**은 민감한 아날로그 RF 회로와 고속 디지털 회로의 성능을 보장하는 기본 전제입니다.
  - 문제 분석: 전원 레일에 포함된 노이즈는 RF 회로의 위상 잡음(phase noise)을 악화시키거나 스푸리어스(spurious) 신호를 발생시켜 통신 품질을 저하시킬 수 있습니다. 또한, 디지털 회로에서는 전압 강하가 타이밍 오류를 유발하여 시스템 오작동의 원인이 될 수 있습니다.<sup>8</sup>
  - 구현 방안: **RF 프론트엔드와 같은 민감한 아날로그 회로에는 저잡음 LDO(Low-Dropout) 레귤레이터를 사용**하고, 각 IC의 전원 핀 가까이 적절한 용량의 디커플링 커패시터를 배치하여 고주파 노이즈를 제거해야 합니다. 또한, PCB의 전원 및 접지면을 넓게 설계하여 임피던스를 낮추고, 고품질의 전원 부품을 선정하여 장기적인 안정성을 확보해야 합니다.
- 펌웨어/게이트웨어의 견고성:
  - 문제 분석: **현장에서 펌웨어나 FPGA 게이트웨어 업데이트 도중 전원 차단 등의 예기치 않은 문제로 인해 이미지가 손상되면, 장비가 부팅되지 않는 '벽돌(brick)' 상태에 빠질 수 있습니다. 이 경우, 물리적인 개입(예: JTAG을 이용한 강제 복구) 없이는 장비를 되살릴 수 없게 되어 막대한 유지보수 비용을 초래합니다.**
  - 구현 방안: **'골든 이미지(Golden Image)' 또는 '폴백 이미지(Fallback Image)' 개념을 도입하는 것이 매우 효과적인 DFR 기법입니다. 이는 검증된 최소 기능의 안정적인 이미지를 플래시 메모리의 보호된 영역에 저장해두는 방식**입니다. 평상시에는 주요 기능을 담은 '사용자 이미지(User Image)'로 부팅하지만, 만약 사용자 이미지 업데이트 후 부팅에 실패하면 부트로더가 이를 감지하고 자동으로 골든 이미지로 부팅을 시도합니다. 골든 이미지로 부팅된 장비는 최소한의 통신 기능을 유지하므로, 원격으로 다시 정상적인 사용자 이미지를 업데이트할 기회를 제공합니다. 이 기능은 재구성이 가능한 SDR 장비에 특히 적합하며, 치명적인 현장 실패를 방지하는 강력한 안전장치 역할을 합니다.<sup>26</sup>

## 기대 효과

DFR 원칙의 체계적인 적용은 시스템의 평균 무고장 시간(MTBF)을 크게 향상시키고, 최대 부하 조건에서도 안정적인 성능을 보장합니다. 또한, 치명적인 현장 고장을 예방하여 유지보수 비용을 절감하고, 제품의 신뢰성에 대한 시장의 평판을 높이는 데 결정적인 기여를 합니다.



## 2.4. 서비스 및 모듈화 용이성 설계 (DFS): 진화를 위한 엔지니어링

DFS(Design for Serviceability)는 제품을 쉽게 유지보수, 수리, 업그레이드할 수 있도록 설계하는 것입니다.<sup>25</sup> 이는 시스템의 다운타임을 최소화하고 제품의 유효 수명을 연장하는 데 중점을 둡니다.

### SDR 특화 DFS 적용 방안

SDR의 DFS는 물리적 측면과 소프트웨어적 측면 모두에서 고려되어야 합니다.

- 물리적 모듈성: RF 프론트엔드를 별도의 도터보드로 구현하는 것은 물리적 모듈성의 대표적인 예입니다.<sup>19</sup> 이를 통해 사용자는 전체 시스템을 교체하지 않고 도터보드만 교체하여 새로운 주파수 대역을 지원하거나 RF 성능을 개선할 수 있습니다. 표준화된 커넥터, 쉽게 접근 가능한 패스너, 그리고 주요 부품의 교체가 용이하도록 내부 공간을 설계하는 것 또한 중요한 물리적 DFS 요소입니다.
- 소프트웨어 모듈성: 이것이 SDR의 DFS의 핵심입니다.
  - 문제 분석: 모든 기능이 하나의 거대한 프로그램으로 묶여 있는 모놀리식(monolithic) 소프트웨어는 특정 기능의 버그를 수정하거나 성능을 개선하기 위해 전체 프로그램을 다시 컴파일하고 배포해야 하므로 유지보수가 매우 비효율적입니다.
  - 구현 방안: GNU Radio와 같은 프레임워크는 소프트웨어 모듈성의 좋은 예를 보여줍니다. GNU Radio에서 하나의 파형(waveform)은 독립적인 신호 처리 블록들의 연결인 '플로우그래프(flowgraph)'로 표현됩니다.<sup>29</sup> 만약 필터 알고리즘을 개선해야 한다면, 전체 애플리케이션이 아닌 해당 필터 블록만 교체하면 됩니다. 또한, USRP 하드웨어 드라이버(UHD) 아키텍처는 호스트 드라이버, 임베디드 장치에서 실행되는 MPM(Module Peripheral Manager) 데몬, 그리고 FPGA 로직을 명확히 분리하여 각 구성 요소를 독립적으로 업데이트하고 관리할 수 있도록 설계되었습니다.<sup>31</sup> 이러한 모듈식 아키텍처는 유지보수 효율성을 극대화합니다.
- 무선/원격 업데이트 (OTA/Remote Updates): SDR의 DFS를 가장 잘 보여주는 기능입니다.
  - 가치 분석: 펌웨어, FPGA 게이트웨어, 그리고 애플리케이션 소프트웨어를 원격으로 안전하게 업데이트할 수 있는 능력은 현장에 배포된 시스템의 가치를 혁신적으로 높입니다.<sup>7</sup> 물리적 접근이 어렵거나 불가능한 환경(예: 위성, 외딴 지역의 기지국)에 설치된 장비의 기능을 개선하고 보안 취약점을 패치하는 데 드는 막대한 비용과 시간을 절약할 수 있습니다.
  - 하드웨어적 지원: 안정적인 원격 업데이트를 위해서는 하드웨어의 지원이 필수적입니다. DFR에서 언급된 골든 이미지 기능을 제공하는 견고한 부트로더, 업데이트 파일을 저장할 충분한 용량의 비휘발성 메모리, 그리고 업데이트 과정 중 예기치 않은 전원 차단으로부터 시스템을 보호하는 전원 관리 시스템이 반드시 필요합니다.

## 기대 효과

DFS를 통해 SDR 시스템은 지원 및 유지보수 비용을 획기적으로 절감하고, 제품의 생애주기를 연장하며, 새로운 기능과 보안 패치를 신속하게 배포할 수 있습니다. 이는 궁극적으로 고객 만족도를 높이고 지속적인 수익 창출의 기반을 마련합니다.

이상의 분석을 통해 SDR 시스템 설계에서 DFX 원칙들이 어떻게 계층적으로 상호작용하며, 하드웨어와 소프트웨어의 경계를 넘어 통합적으로 적용되어야 하는지가 명확해집니다. DFM/DFA는 가장 기본적인 물리적 토대를 마련하고, DFT는 이 토대가 제대로 구축되었는지 검증하는 도구를 제공합니다. DFR은 검증된 제품이 실제 환경에서 안정적으로 작동하도록 보장하며, 마지막으로 DFS는 안정적인 제품이 시간의 흐름에 따라 변화하고 진화할 수 있는 능력을 부여합니다. 특히 SDR에서는 하드웨어 설계가 단순히 물리적 특성을 결정하는 것을 넘어, 신뢰성 있는 원격 업데이트나 원격 진단과 같은 소프트웨어 기반의 생애주기 관리 기능을 '가능하게 하는' 핵심 조력자 역할을 수행한다는 점을 이해하는 것이 매우 중요합니다.

## 섹션 3: 실용적 구현 가이드: 튜토리얼 접근 방식

이론적 논의를 넘어, 이 섹션에서는 실제 엔지니어링 현장에서 DFX 원칙을 적용하는 구체적인 방법을 튜토리얼 형식으로 제시합니다. 이를 위해 널리 사용되는 오픈소스 SDR 플랫폼을 사례 연구 대상으로 삼아, 실제 설계 자료를 바탕으로 실질적인 가이드를 제공합니다.

### 3.1. 사례 연구 플랫폼: 오픈소스 SDR 하드웨어 개요

본 튜토리얼에서는 두 가지 대표적인 오픈소스 SDR 플랫폼, 즉 Ettus Research의 USRP 시리즈와 Myriad-RF의 LimeSDR 시리즈를 활용합니다. 이 플랫폼들은 높은 인지도, 회로도 및 소스 코드의 공개, 그리고 각각 뚜렷한 설계 철학(USRP: 모듈성 및 전문가용, LimeSDR: 고집적 및 비용 효율성)을 가지고 있어 DFX 원칙을 설명하기에 이상적인 사례입니다.

- **Ettus Research USRP:** 메인보드와 RF 도터보드가 분리된 모듈식 아키텍처와 전문적인 수준의 UHD(USRP Hardware Driver) 소프트웨어 아키텍처가 특징입니다. 이는 DFS와 물리적 모듈성의 좋은 예시를 제공합니다.<sup>31</sup>
- **Myriad-RF LimeSDR:** 고도로 집적된 LMS7002M RF 트랜시버 칩을 중심으로 설계되어 비용 효율성이 높으며, 강력한 커뮤니티 지원을 받습니다. 특히, 회로도, PCB 레이아웃,

제조 파일까지 모두 GitHub를 통해 공개되어 있어 DFM과 DFT를 분석하기에 매우 적합합니다.<sup>38</sup>

다음 표는 이들 오픈소스 SDR 플랫폼의 주요 모델들이 DFT 및 DFS 관련 기능을 어떻게 구현하고 있는지를 비교하여 보여줍니다. 이는 이어지는 튜토리얼의 배경 지식을 제공하고, 각 설계의 장단점을 파악하는 데 도움을 줍니다.

표 2: 오픈소스 SDR 플랫폼의 DFT/DFS 기능 비교

SDR 모델	JTAG 접근	시리얼 콘솔 접근	원격 관리 API	FPGA 멀티부트/폴백	물리적 모듈성
USRP X310	온보드 JTAG 프로그래머 <sup>19</sup>	후면 패널 시리얼 포트	UHD API	플래시 이미지 로더 지원 <sup>19</sup>	도터보드 (2개 슬롯)
USRP E310	USB를 통한 내장 JTAG 디버거 <sup>24</sup>	USB를 통한 시리얼 콘솔 <sup>24</sup>	UHD MPM (임베디드)	SD 카드 기반 부팅	일체형
LimeSDR-USB	미실장 패드	-	LimeSuite API	-	일체형
LimeSDR Mini v2	미실장 패드 <sup>42</sup>	-	LimeSuite API	듀얼부트 지원 <sup>27</sup>	일체형
LimeSDR-XTRX	PCIe 어댑터 또는 핀헤더 <sup>26</sup>	-	LimeSuite API	골든/사용자 이미지 멀티부트 <sup>26</sup>	Mini-PCIe 폼팩터

## 3.2. 튜토리얼: SDR 하드웨어 설계에 DFM 및 DFT 구현하기

### 3.2.1. LimeSDR Mini v2 회로도 분석

- 목표: 엔지니어가 DFX 관점에서 회로도를 해석하는 방법을 학습합니다.
- 절차: 공개된 LimeSDR Mini v2 하드웨어 설계 파일을 사용하여 회로도의 주요 부분을 단계별로 분석합니다.<sup>39</sup>
  1. **DFM 사례 (전원부):** 회로도의 전원부 섹션을 살펴봅니다. 여러 개의 저잡음 LDO(LD39100PUR 등)와 스위칭 레귤레이터(LMZ20501)를 사용하여 1.1V, 1.8V, 3.3V 등 다양한 전압을 생성하는 것을 확인할 수 있습니다.<sup>42</sup> 이 부품들은 널리 사용되는 표준 부품으로, 안정적인 공급과 비용 효율성을 고려한 DFM의 좋은 예입니다. 각 전원 레일이 모듈화되어 있어 문제 발생 시 디버깅이 용이합니다.
  2. **DFT 사례 (JTAG 인터페이스):** FPGA 주변 회로를 보면 'FPGA\_JTAG'이라는 이름의 커넥터(J5)가 미실장(unpopulated) 상태로 설계되어 있음을 알 수 있습니다.<sup>42</sup> 이는 대량 생산 시에는 비용 절감을 위해 커넥터를 실장하지 않지만, 개발 및 디버깅, 생산 라인에서의 초기 테스트를 위해 JTAG 인터페이스를 사용할 수 있도록 설계한 것입니다. 이 핀헤더는 FPGA의 JTAG 핀(TCK, TMS, TDI, TDO)에 직접 연결되어 있습니다.
  3. **DFT 사례 (테스트 포인트):** PCB 레이아웃 파일을 살펴보면, 주요 전압 레일(VCC\_1V1, VCC\_3V3 등)과 클럭 신호 경로에 작은 원형 패드 형태의 테스트 포인트가 배치되어 있는 것을 발견할 수 있습니다. 이는 생산 테스트나 수리 과정에서 프로브를 쉽게 접촉하여 전압과 신호를 측정할 수 있도록 한 전형적인 DFT 기법입니다.

### 3.2.2. JTAG 디버그 인터페이스 추가 단계별 가이드

- 목표: 핵심적인 DFT 기능인 JTAG 인터페이스를 실제 설계에 추가하는 실용적인 방법을 제공합니다.
- 절차:
  1. **JTAG 신호 이해:** JTAG의 4가지 필수 신호(TDI, TDO, TCK, TMS)와 선택적 신호(TRST)의 역할을 설명합니다. 각 신호는 테스트 데이터를 직렬로 입력하고, 출력하며, 클럭을 공급하고, 테스트 모드를 선택하는 기능을 수행합니다.<sup>22</sup>
  2. 참조 회로도 제시: 아래와 같이 표준 2x5핀 0.05인치 피치 JTAG 헤더를 위한 간단한 참조 회로도를 제시합니다. 업계 모범 사례에 따라 TCK, TMS, TDI 신호에는 풀업(Pull-up) 저항을, TDO에는 풀다운(Pull-down) 저항을 추가하여 신호 안정성을 확보하는 것이 좋습니다.

[FPGA]

1 (VREF) ----+---[1k]---+--- 3.3V

|

2 (TMS) ----+-----> FPGA\_TMS

|

3 (GND) ----+-----> GND

|

```

4 (TCK) ----+-----> FPGA_TCK
|
5 (GND) ----+-----> GND
|
6 (TDO) ----+--[10k]---+< FPGA_TDO
||
7 (NC) | +-- GND
|
8 (TDI) ----+-----> FPGA_TDI
|
9 (GND) ----+-----> GND
|
10 (nTRST)---+-----> FPGA_nTRST (Optional)
...

```

3. **\*\*PCB 레이아웃 시연:\*\*** FPGA 핀에서 JTAG 헤더까지 신호를 배선하는 개념적인 PCB 레이아웃 예시를 보여줍니다. TCK와 같은 고속 신호는 다른 신호와의 간섭을 피하기 위해 가능한 짧고 직선으로 배선하고, 주변에 접지면을 확보하여 신호 무결성을 유지해야 함을 강조합니다.

4. **\*\*소프트웨어 연결:\*\*** 이 JTAG 헤더가 Xilinx Platform Cable USB와 같은 JTAG 프로그래머와 연결되고, Vivado의 Hardware Manager와 같은 소프트웨어 도구를 통해 FPGA를 인식하고 디버깅하는 과정을 간략하게 설명합니다.[37]

### 3.3. 튜토리얼: FPGA 폴백 이미지를 통한 DFR 구현

- **목표:** SDR에 특화된 강력한 신뢰성 기능인 폴백 이미지(fallback image) 구현 방법을 상세히 안내합니다.
- **절차:** 이 튜토리얼은 LimeSDR-XTRX 게이트웨어 저장소의 문서를 기반으로 구성됩니다.<sup>26</sup>
  1. **개념 설명:** '사용자(user)' 이미지와 '골든(gold)' 이미지의 패러다임을 설명합니다. '골든' 이미지는 사용자가 업데이트한 '사용자' 이미지가 손상되어 부팅에 실패할 경우 시스템을 복구할 수 있는 최소 기능의 안전장치 역할을 합니다.
  2. **부트 로직 상세 설명:** 멀티부트(multiboot) 프로세스를 설명합니다. 전원이 켜지면 FPGA는 먼저 플래시 메모리의 '사용자' 이미지 주소에서 이미지를 로드하려고 시도합니다. 만약 이 과정에서 CRC 오류나 타임아웃 등으로 실패하면, FPGA는 자동으로 미리 지정된 '골든' 이미지 주소에서 이미지를 로드하여 부팅을 재시도합니다.
  3. **복구 절차 제공:** 손상된 사용자 이미지로 인해 '벽돌'이 된 장비를 복구하는 절차를

단계별로 제시합니다. 먼저, JTAG을 사용하여 'combined\_flash\_programming\_file.bin' (골든+사용자 이미지 통합 파일)을 플래시 메모리에 강제로 프로그래밍하여 초기 상태로 되돌립니다. 또는, 골든 이미지로 부팅된 상태에서 LimeSuite 소프트웨어의 '골든 이미지 쓰기 기능'을 사용하여 정상적인 사용자 이미지를 다시 프로그래밍할 수 있습니다. 이는 DFT(JTAG) 기능이 DFR(멀티부트)과 DFS(현장 복구)를 가능하게 하는 상호 의존적인 관계를 명확히 보여줍니다.

4. 파일 구조 제시: 개념을 구체화하기 위해 저장소에 있는 실제 파일명(user\_flash\_programming\_file.bin, gold\_flash\_programming\_file.bin, combined\_flash\_programming\_file.bin)을 명시하여 각 파일의 역할을 설명합니다.

### 3.4. 튜토리얼: 모듈식 소프트웨어를 통한 DFS 아키텍처 설계

- 목표: 소프트웨어 아키텍처 선택이 어떻게 서비스 용이성에 직접적인 영향을 미치는지 보여줍니다.
- 절차:

#### 3.4.1. GNU Radio에서의 모듈식 파형 설계

1. **GNU Radio Companion (GRC) 소개:** GRC를 시각적인 플로우그래프 기반 설계 도구로 소개합니다. 사용자는 코드를 직접 작성하지 않고도 신호 처리 블록을 연결하여 복잡한 SDR 애플리케이션을 만들 수 있습니다.<sup>29</sup>
2. 간단한 **WBFM** 수신기 제작: 간단한 광대역 FM(WBFM) 수신기 플로우그래프를 단계별로 만듭니다. RTL-SDR Source 블록으로 신호를 수신하고, Low Pass Filter 블록으로 불필요한 신호를 제거한 뒤, WBFM Receive 블록으로 복조하고, Audio Sink 블록으로 소리를 출력합니다.
3. 모듈성 강조: 이 플로우그래프에서 각 블록은 독립적인 모듈임을 설명합니다. 만약 더 효율적인 저역 통과 필터 알고리즘이 개발된다면, 전체 애플리케이션을 재설계할 필요 없이 Low Pass Filter 블록만 교체하면 됩니다. 이는 유지보수와 업그레이드를 매우 용이하게 만듭니다.
4. 계층적 블록 소개: 필터와 복조기 부분을 하나의 'FM Demod'라는 커스텀 블록으로 캡슐화하는 방법을 보여줍니다. 이를 통해 복잡한 시스템을 더 작고, 유지보수하기 쉬우며, 재사용 가능한 모듈들로 구성할 수 있음을 설명합니다. 이는 대규모 SDR 시스템의 서비스 용이성을 높이는 핵심적인 기법입니다.<sup>46</sup>

#### 3.4.2. 원격 관리를 위한 USRP UHD 아키텍처 활용

1. **UHD 아키텍처 설명:** USRP N310, E320과 같은 임베디드 USRP 제품군에서 사용되는 MPM(Module Peripheral Manager) 아키텍처를 설명합니다.<sup>32</sup>
2. 클라이언트-서버 모델: MPM 데몬은 USRP 장비 내부의 임베디드 리눅스에서 실행되는 서버 역할을 하며, 사용자의 호스트 PC에서 실행되는 애플리케이션(클라이언트)은 네트워크 RPC(Remote Procedure Call)를 통해 MPM

데몬과 통신하여 장비를 제어합니다.

3. 서비스 용이성 강조: 이 아키텍처는 본질적으로 DFS를 지원합니다. 원격 호스트는 주요 신호 처리 애플리케이션을 실행하지 않고도 MPM 데몬에 접속하여 장치의 상태를 조회하고, 센서 값을 읽으며(DFT), 펌웨어나 FPGA 업데이트를 트리거할 수 있습니다. 이처럼 제어/관리 평면(Control/Management Plane)을 데이터 평면(Data Plane)과 분리하는 것은 현대적인 서비스 지향 시스템의 핵심적인 설계 원칙입니다.

이러한 튜토리얼들을 통해 명확해지는 사실은, DFX 원칙들이 서로 긴밀하게 연결되어 있다는 점입니다. DFR을 위한 멀티부트 기능은 DFT 기능인 JTAG 없이는 완벽하게 구현되거나 복구되기 어렵습니다. 또한, 오픈소스 SDR 프로젝트의 존재는 이러한 모범 사례를 학습하고 전파하는 데 결정적인 역할을 합니다. 공개된 회로도와 소스 코드는 DFX 원칙이 실제 제품에 어떻게 구현되는지를 보여주는 살아있는 교과서이며, 커뮤니티 전체가 이를 통해 배우고 발전할 수 있는 기반을 제공합니다.

## 섹션 4: 종합 및 전략적 제언

본 보고서의 분석을 종합하여, SDR 개발을 주도하는 엔지니어링 리더들을 위한 전략적 제언을 제시합니다. 이는 DFX를 단편적인 기술이 아닌, 조직의 핵심 역량으로 내재화하기 위한 지침이 될 것입니다.

### 4.1. SDR 개발 워크플로우에 DFX 통합: 단계별 접근법

DFX는 개발 마지막 단계에서 확인하는 체크리스트가 아니라, 프로젝트 첫날부터 모든 의사결정에 스며들어야 하는 철학입니다.<sup>5</sup> 이를 성공적으로 정착시키기 위해서는 단계별 접근이 효과적입니다.

- **개념 단계 (Concept Phase):** DFX는 여기서 시작됩니다. 프로젝트의 목표를 정의할 때부터 다음과 같은 DFX 관련 질문을 반드시 포함해야 합니다.
  - 목표 생산량은 얼마인가? (DFM에 직접적인 영향을 미침)
  - 요구되는 시스템 가동 시간과 신뢰성 수준은 어느 정도인가? (DFR의 목표를 설정함)
  - 예상되는 현장 운용 기간과 업그레이드 전략은 무엇인가? (DFS의 범위를 결정함)
- **설계 단계 (Design Phase):** 하드웨어, 소프트웨어, 제조, 테스트 엔지니어가 모두 참여하는 교차 기능 설계 검토(Cross-functional Design Review) 팀을 구성하고 정기적으로 운영해야 합니다. 이 과정에서 각 DFX 원칙에 대한 체크리스트를 활용하여 잠재적인 문제점을 체계적으로 식별하고 설계에 반영합니다. 예를 들어, "모든 BGA 부품은 AXI 검사가 가능한가?", "주요 전원 레일에 대한 테스트 포인트가 배치되었는가?", "FPGA의 예상 발열량에 대한 열 시뮬레이션 결과는 충분한 마진을 확보했는가?" 와 같은 구체적인

항목을 검토해야 합니다.

- **프로토타입 단계 (Prototyping Phase):** 설계에 반영된 DFX 기능들을 실제로 테스트하고 검증하는 것이 매우 중요합니다. 프로토타입 보드가 제작되면, JTAG 포트가 정상적으로 동작하는지, 계획된 원격 업데이트 시나리오가 실패 없이 수행되는지, 의도적으로 손상된 이미지를 주입했을 때 폴백 기능이 정상적으로 작동하여 시스템을 복구할 수 있는지 등을 반드시 시험해야 합니다. 이 단계에서 발견된 문제점은 양산 단계 이전에 수정되어야 막대한 비용 손실을 막을 수 있습니다.

## 4.2. 미래 전망: 차세대 SDR에서 DFX의 역할

SDR 기술이 계속해서 발전함에 따라 DFX의 중요성은 더욱 커질 것입니다.

- **인지 무선 (Cognitive Radio):** SDR이 주변 전파 환경을 스스로 감지하고 최적의 통신 방식을 자율적으로 선택하는 인지 무선 기술에서는, 시스템의 안정성과 복원력이 무엇보다 중요합니다. 자율적으로 동작하는 라디오가 안정적으로 업데이트되지 못하거나 치명적인 오류로부터 스스로 복구할 수 없다면, 이는 신뢰할 수 없는 자산이 될 뿐입니다. 따라서 견고한 DFR과 원격 DFS 기능은 인지 무선 기술의 실용화를 위한 핵심 전제 조건이 될 것입니다.
- **엣지 AI/ML (AI/ML at the Edge):** SDR은 네트워크 엣지에서 AI/머신러닝 추론을 수행하는 플랫폼으로 점점 더 많이 활용되고 있습니다. 이는 하드웨어에 막대한 연산 부하를 가하며, 필연적으로 높은 전력 소비와 발열을 동반합니다. 따라서 차세대 AI 기반 SDR에서는 DFR, 특히 효율적인 열 관리가 시스템의 성능과 수명을 결정하는 가장 중요한 설계 동인이 될 것입니다.

결론적으로, SDR이 더욱 강력해지고 자율화될수록, 그 소프트웨어 정의 능력은 DFX 방법론을 통해 구축된 견고하고, 신뢰할 수 있으며, 서비스 가능한 하드웨어 기반에 더욱 깊이 의존하게 될 것입니다. DFX는 미래의 라디오가 '소프트웨어로 정의'되면서도 동시에 '현실 세계에서 검증'될 수 있도록 보장하는 핵심적인 규율입니다. 이는 기술적 우위를 넘어, 지속 가능한 비즈니스 가치를 창출하는 가장 확실한 길입니다.

### 참고 자료

1. itpe.jackerlab.com, 9월 27, 2025에 액세스, [https://itpe.jackerlab.com/entry/SDRSoftware-Defined-Radio#:~:text=SDR\(Software%20Defined%20Radio\)%EC%9D%80,%EC%84%A4%EA%B3%84%EB%90%9C%20%EB%AC%B4%EC%84%A0%20%ED%86%B5%EC%8B%A0%20%EA%B8%B0%EC%88%A0%EC%9E%85%EB%8B%88%EB%8B%A4](https://itpe.jackerlab.com/entry/SDRSoftware-Defined-Radio#:~:text=SDR(Software%20Defined%20Radio)%EC%9D%80,%EC%84%A4%EA%B3%84%EB%90%9C%20%EB%AC%B4%EC%84%A0%20%ED%86%B5%EC%8B%A0%20%EA%B8%B0%EC%88%A0%EC%9E%85%EB%8B%88%EB%8B%A4).
2. 소프트웨어 정의 라디오 - 위키백과, 우리 모두의 백과사전, 9월 27, 2025에 액세스, [https://ko.wikipedia.org/wiki/%EC%86%8C%ED%94%84%ED%8A%B8%EC%9B%A8%EC%96%B4\\_%EC%A0%95%EC%9D%98\\_%EB%9D%BC%EB%94%94%EC%98%A4](https://ko.wikipedia.org/wiki/%EC%86%8C%ED%94%84%ED%8A%B8%EC%9B%A8%EC%96%B4_%EC%A0%95%EC%9D%98_%EB%9D%BC%EB%94%94%EC%98%A4)



3. SDR (Software Defined Radio): 하드웨어 중심 무선 통신의 혁신적 소프트웨어 전환 기술, 9월 27, 2025에 액세스,  
<https://rupijun.tistory.com/m/entry/SDR-Software-Defined-Radio-%ED%95%98%B%93%9C%EC%9B%A8%EC%96%B4-%EC%A4%91%EC%8B%AC-%EB%AC%B4%EC%84%A0-%ED%86%B5%EC%8B%A0%EC%9D%98-%ED%98%81%EC%8B%A0%EC%A0%81-%EC%86%8C%ED%94%84%ED%8A%B8%EC%9B%A8%EC%96%B4-%EC%A0%84%ED%99%98-%EA%B8%B0%EC%88%A0>
4. What is DFX? - Boston Engineering, 9월 27, 2025에 액세스,  
<https://blog.boston-engineering.com/what-is-dfx>
5. What is Design for Excellence (DfX)? - Ansys, 9월 27, 2025에 액세스,  
<https://www.ansys.com/blog/what-is-dfx>
6. 차세대 단말기를 위한 SDR 기반의 통신모뎀 구조, 9월 27, 2025에 액세스,  
<https://tta.or.kr/data/androReport/ttaJnal/9-5.pdf>
7. SDR(Software Defined Radio) - ITPE \* JackerLab, 9월 27, 2025에 액세스,  
<https://itpe.jackerlab.com/entry/SDRSoftware-Defined-Radio>
8. Software defined radio design resources | TI.com - Texas Instruments, 9월 27, 2025에 액세스, <https://www.ti.com/solution/software-defined-radio>
9. SDR(Software Defined Radio) - Korea Science, 9월 27, 2025에 액세스,  
<https://koreascience.kr/article/JAKO200270709946689.pdf>
10. Software Defined Radios - Architectures, Systems and Functions - NASA Technical Reports Server (NTRS), 9월 27, 2025에 액세스,  
<https://ntrs.nasa.gov/citations/20170005339>
11. Software-defined Radios: Architecture, State-of-the-art, and Challenges - arXiv, 9월 27, 2025에 액세스, <https://arxiv.org/pdf/1804.06564>
12. Designing Software Defined Radio (SDR) with RFSoc System on Module, 9월 27, 2025에 액세스,  
<https://www.electronicmedia.info/2025/07/28/designing-software-defined-radio-sdr-with-rfsoc-system-on-module/>
13. Design-for-Reliability on Thermal Management and ESD Protection of Integrated Circuits, 9월 27, 2025에 액세스, <https://escholarship.org/uc/item/6tm0c72g>
14. What Is Design for Manufacturing (DFM)? - Studio Red, 9월 27, 2025에 액세스,  
<https://www.studiored.com/blog/design/design-for-manufacturing/>
15. Design for manufacturing output files - optimizing engineering to manufacturing efficiency - Electronic Systems Design - Siemens Digital Industries Software Blogs, 9월 27, 2025에 액세스,  
<https://blogs.sw.siemens.com/electronic-systems-design/2025/03/18/design-for-manufacturing-output-files/>
16. Design for Manufacturing (DFM): A Complete Guide - DISHER, 9월 27, 2025에 액세스, <https://www.disher.com/blog/design-for-manufacturing/>
17. Design for Manufacturing (DFM) Best Practices When Designing Medical Device Products, 9월 27, 2025에 액세스,  
<https://arrotek.com/design-for-manufacturing-dfm-best-practices-when-designing-medical-device-products/>
18. Design for Manufacturing (DFM) - Semiconductor Engineering, 9월 27, 2025에 액세스,

- [https://semiengineering.com/knowledge\\_centers/eda-design/methodologies-and-flows/design-for-manufacturing-dfm/](https://semiengineering.com/knowledge_centers/eda-design/methodologies-and-flows/design-for-manufacturing-dfm/)
19. USRP Hardware Driver and USRP Manual: USRP X3x0 Series - Ettus Research, 9월 27, 2025에 액세스, [https://files.ettus.com/manual/page\\_usrp\\_x3x0.html](https://files.ettus.com/manual/page_usrp_x3x0.html)
  20. Design for Testability methodologies applied to a RISC-V processor - UPCCommons, 9월 27, 2025에 액세스, <https://upcommons.upc.edu/bitstreams/b681ca58-01f6-4fa0-8f31-caaef5551710/download>
  21. 20. Design for Testability - Computer Engineering Research Center (CERC) - University of Texas at Austin, 9월 27, 2025에 액세스, <https://www.cerc.utexas.edu/~jaa/vlsi/lectures/20-2.pdf>
  22. JTAG - Wikipedia, 9월 27, 2025에 액세스, <https://en.wikipedia.org/wiki/JTAG>
  23. JTAG Boundary Scan Tutorial - Etoolsmiths, 9월 27, 2025에 액세스, <https://www.etoolsmiths.com/jtag-boundary-scan-tutorial/>
  24. USRP E3xx Series, 9월 27, 2025에 액세스, [https://uhd.readthedocs.io/en/latest/page\\_usrp\\_e3xx.html](https://uhd.readthedocs.io/en/latest/page_usrp_e3xx.html)
  25. Design for Serviceability/Maintainability and the Strategy Beyond the Purchase - Boston Engineering, 9월 27, 2025에 액세스, <https://blog.boston-engineering.com/design-for-serviceability-maintainability-and-the-strategy-beyond-the-purchase>
  26. myriardf/LimeSDR-XTRX\_GW: LimeSDR XTRX gateway ... - GitHub, 9월 27, 2025에 액세스, [https://github.com/myriardf/LimeSDR-XTRX\\_GW](https://github.com/myriardf/LimeSDR-XTRX_GW)
  27. LimeSDR Mini v2 gateway project - GitHub, 9월 27, 2025에 액세스, [https://github.com/myriardf/LimeSDR-Mini-v2\\_GW](https://github.com/myriardf/LimeSDR-Mini-v2_GW)
  28. DFS : Design for Serviceability - Super Engineer, 9월 27, 2025에 액세스, <https://www.superengineer.net/blog/dfx-dfs-design-for-serviceability>
  29. What Is GNU Radio - GNU Radio, 9월 27, 2025에 액세스, [https://wiki.gnuradio.org/index.php?title=What\\_Is\\_GNU\\_Radio](https://wiki.gnuradio.org/index.php?title=What_Is_GNU_Radio)
  30. Review of Design and Implementation of New Software Radio System based on GNU Radio and USRP - ResearchGate, 9월 27, 2025에 액세스, [https://www.researchgate.net/publication/390960689\\_Review\\_of\\_Design\\_and\\_Implementation\\_of\\_New\\_Software\\_Radio\\_System\\_based\\_on\\_GNU\\_Radio\\_and\\_USRP](https://www.researchgate.net/publication/390960689_Review_of_Design_and_Implementation_of_New_Software_Radio_System_based_on_GNU_Radio_and_USRP)
  31. EttusResearch/uhd: The USRP™ Hardware Driver Repository - GitHub, 9월 27, 2025에 액세스, <https://github.com/EttusResearch/uhd>
  32. The Module Peripheral Manager (MPM) Architecture - USRP Hardware Driver and USRP Manual, 9월 27, 2025에 액세스, [https://uhd.readthedocs.io/en/uhd-4.8/page\\_mpm.html](https://uhd.readthedocs.io/en/uhd-4.8/page_mpm.html)
  33. SDR 소프트웨어 구조 및 다운로드, 9월 27, 2025에 액세스, <https://koreascience.kr/article/JAKO200310102447302.page>
  34. Inside Software Defined Radio Architecture and Waveform Development - Mobilicom, 9월 27, 2025에 액세스, <https://mobilicom.com/insight/inside-sdr-and-waveforms/>
  35. The Advantages and Future Trends of SDR in Unmanned Systems, 9월 27, 2025에 액세스,

<https://www.simpulse-sdr.com/articles/the-advantages-and-future-trends-of-sdr-in-unmanned-systems>

36. Ettus Research - GitHub, 9월 27, 2025에 액세스, <https://github.com/ettusresearch>
37. EttusResearch/fpga: The USRP™ Hardware Driver FPGA ... - GitHub, 9월 27, 2025에 액세스, <https://github.com/EttusResearch/fpga>
38. LimeSDR X3 hardware design. - GitHub, 9월 27, 2025에 액세스, <https://github.com/myriadrf/LimeSDR-X3>
39. myriadrf/LimeSDR-Mini-v2 - GitHub, 9월 27, 2025에 액세스, <https://github.com/myriadrf/LimeSDR-Mini-v2>
40. LimeSDR Support - Lime Microsystems, 9월 27, 2025에 액세스, <https://limemicro.com/support/limesdr/>
41. LimeSDR - Lime Microsystems, 9월 27, 2025에 액세스, <https://limemicro.com/boards/limesdr/>
42. LimeSDR Mini v2.2 Board, 9월 27, 2025에 액세스, <https://limesdr-mini.myriadrf.org/v2.2/>
43. LimeSDR Mini hardware v1.2 - MyriadRF Discourse, 9월 27, 2025에 액세스, <https://discourse.myriadrf.org/t/limesdr-mini-hardware-v1-2/3153>
44. Technical Guide to JTAG - XJTAG Tutorial, 9월 27, 2025에 액세스, <https://www.xjtag.com/about-jtag/jtag-a-technical-overview/>
45. Introduction to SDR and GNU Radio - People, 9월 27, 2025에 액세스, <https://people.ece.ubc.ca/edc/7860.jan2021/lab2.pdf>
46. Tutorials - GNU Radio, 9월 27, 2025에 액세스, <https://wiki.gnuradio.org/index.php/Tutorials>