



소프트웨어공학개론

T6 김선홍 박정빈 송지연

| 피드백 사항 및 질문 사항 정리

| 1. 외부 DB와 내부 DB

- 내부 DB: post에서 사용하는 판매 목록을 위한 data store
- 외부 DB: POS와 외부 상품 재고 서버에 대한 정보를 의미

| 2. interface

- 외부에서 data를 받을 때 사용되는 매개체.
- 반드시 물리적인 매개체로 존재할 필요는 없기에 소프트웨어와 하드 웨어 둘 다 가능

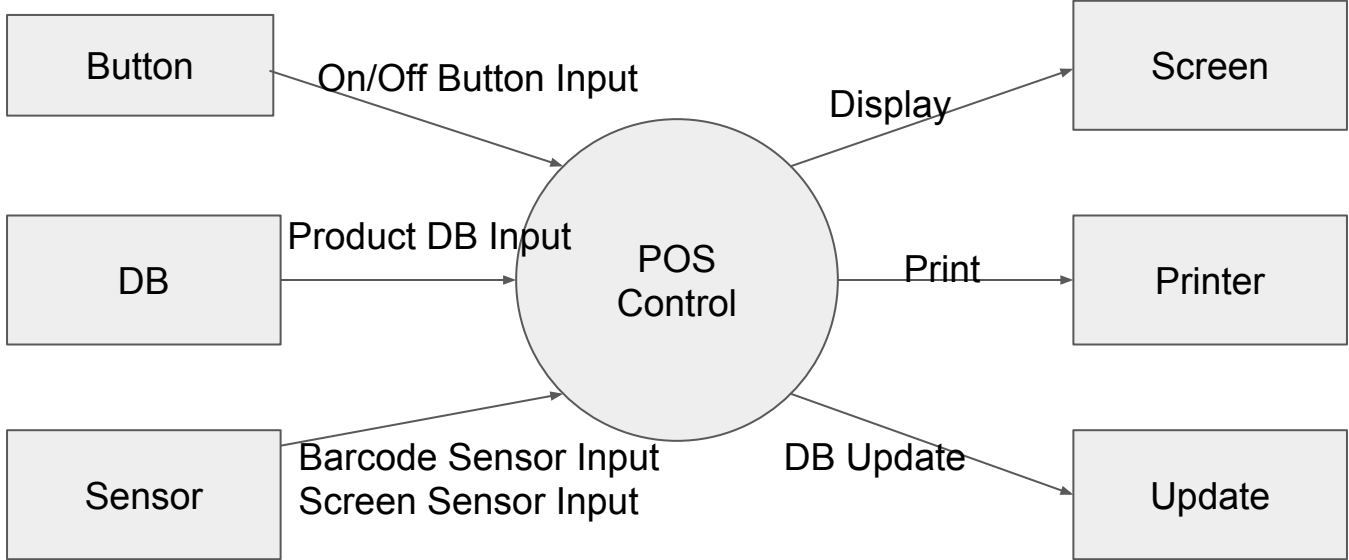
| 피드백 사항 및 질문 사항 정리

| 내부 DB에 대한 구현

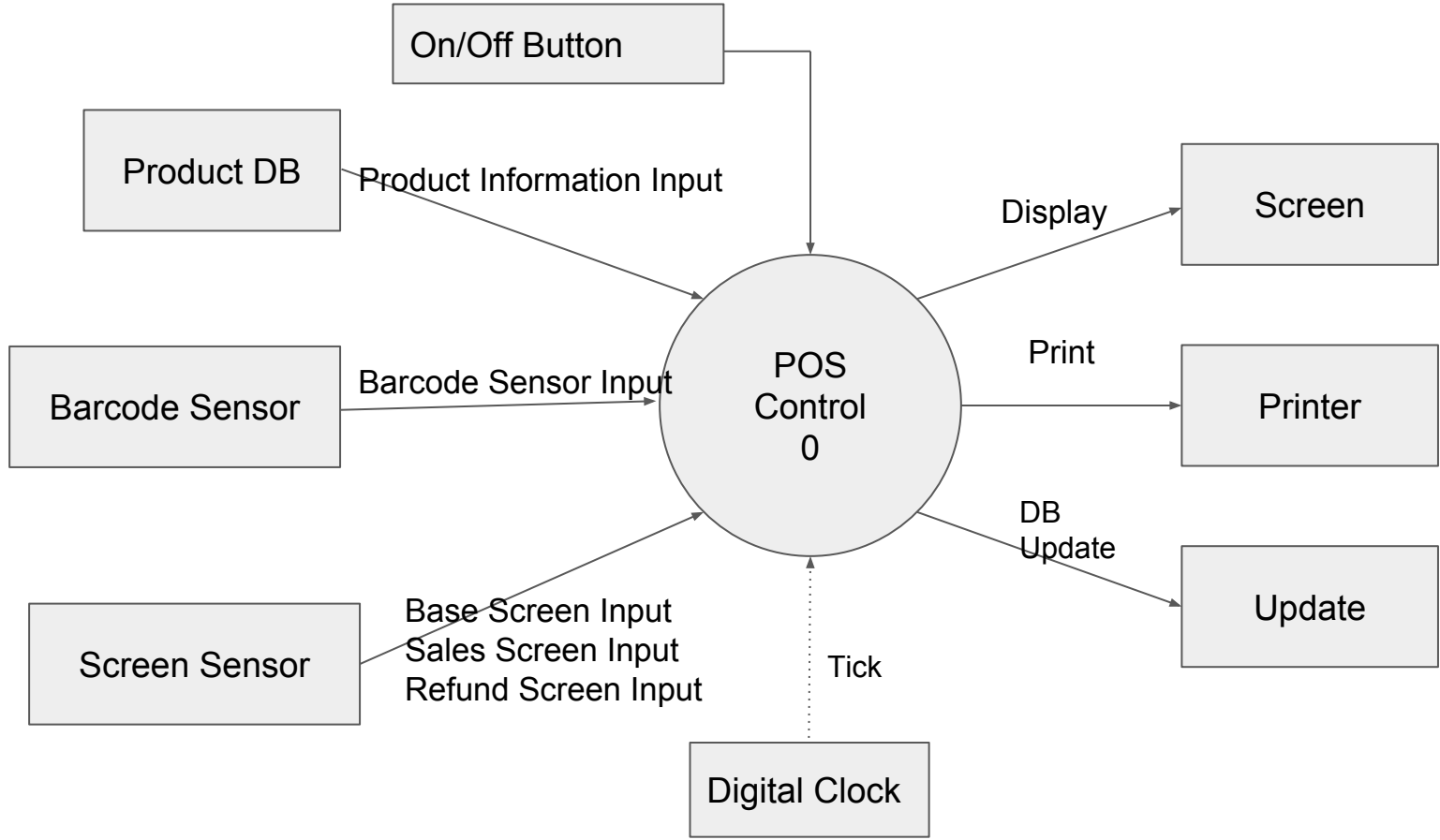
- data store와 관련되어 DFD 상에서 data 추가 예정

| DFD , SD 1.4 interface

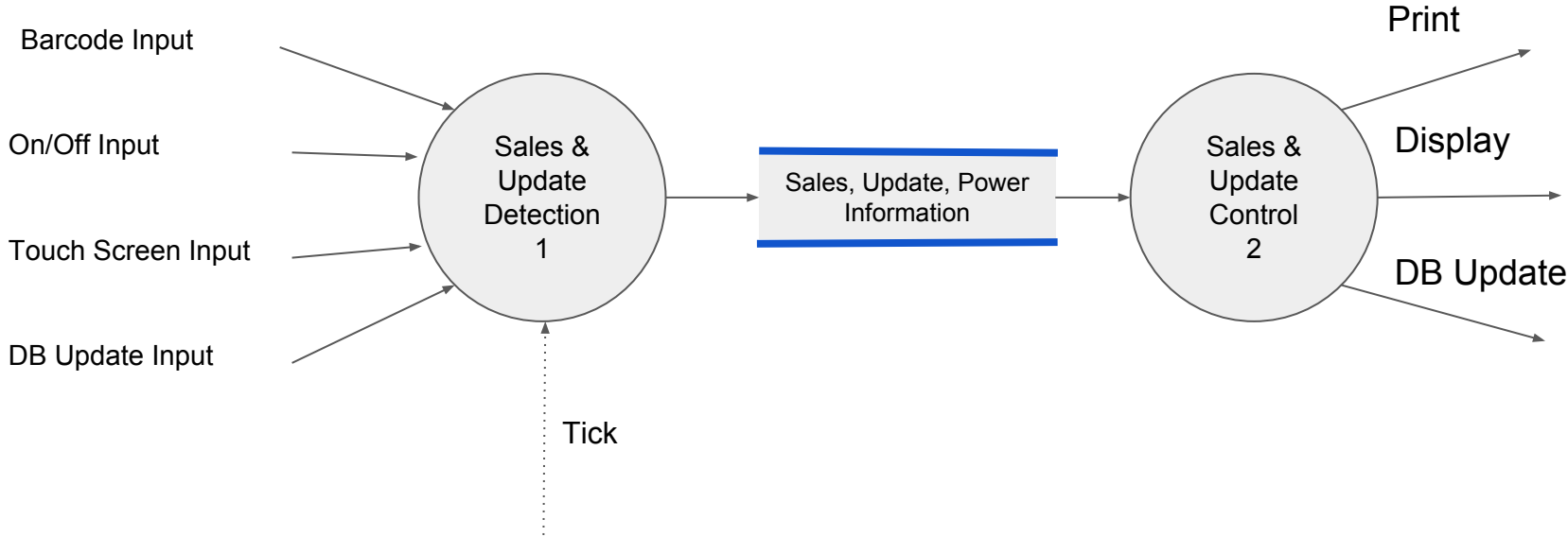
| System Context Diagram



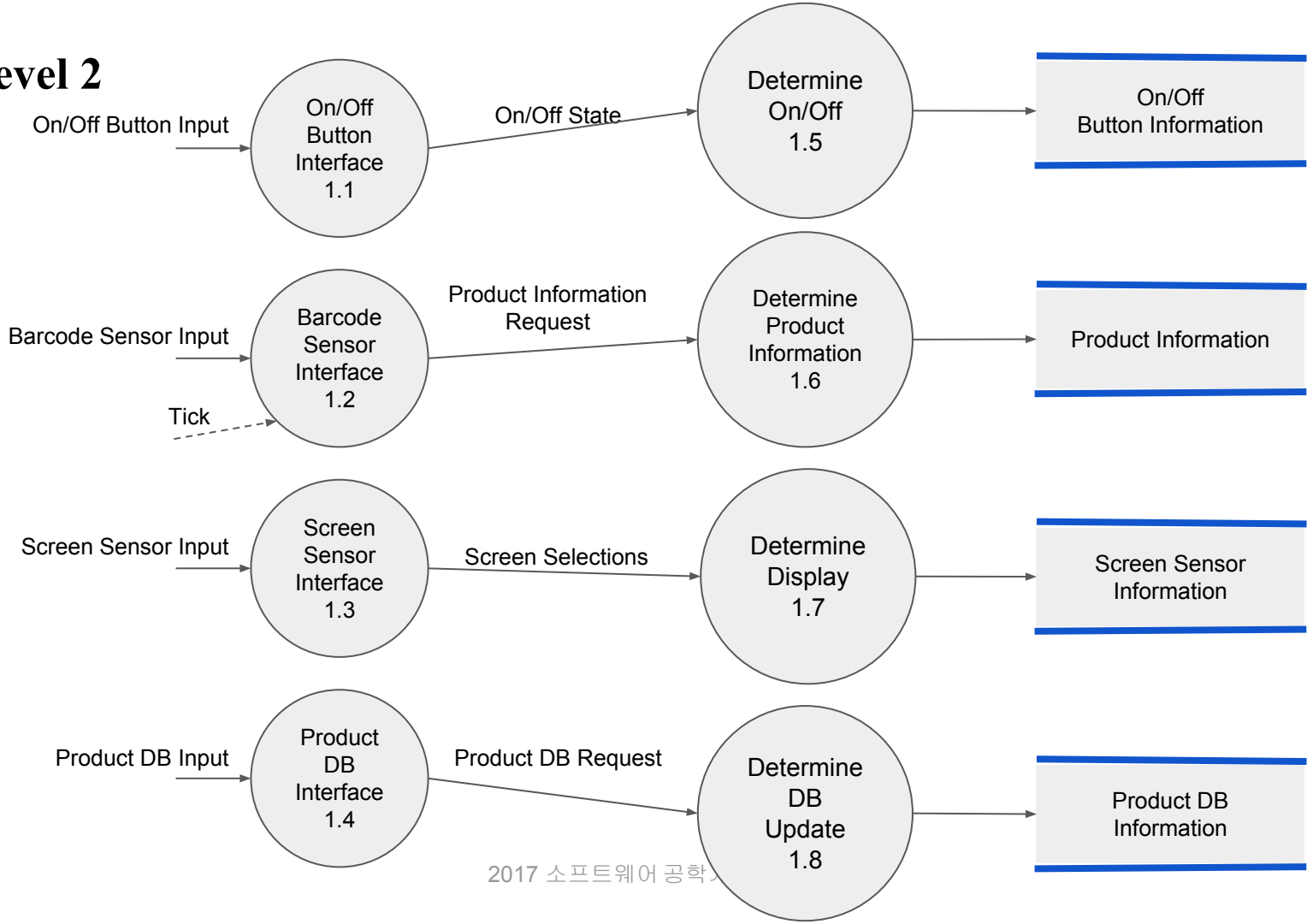
DFD Level 0



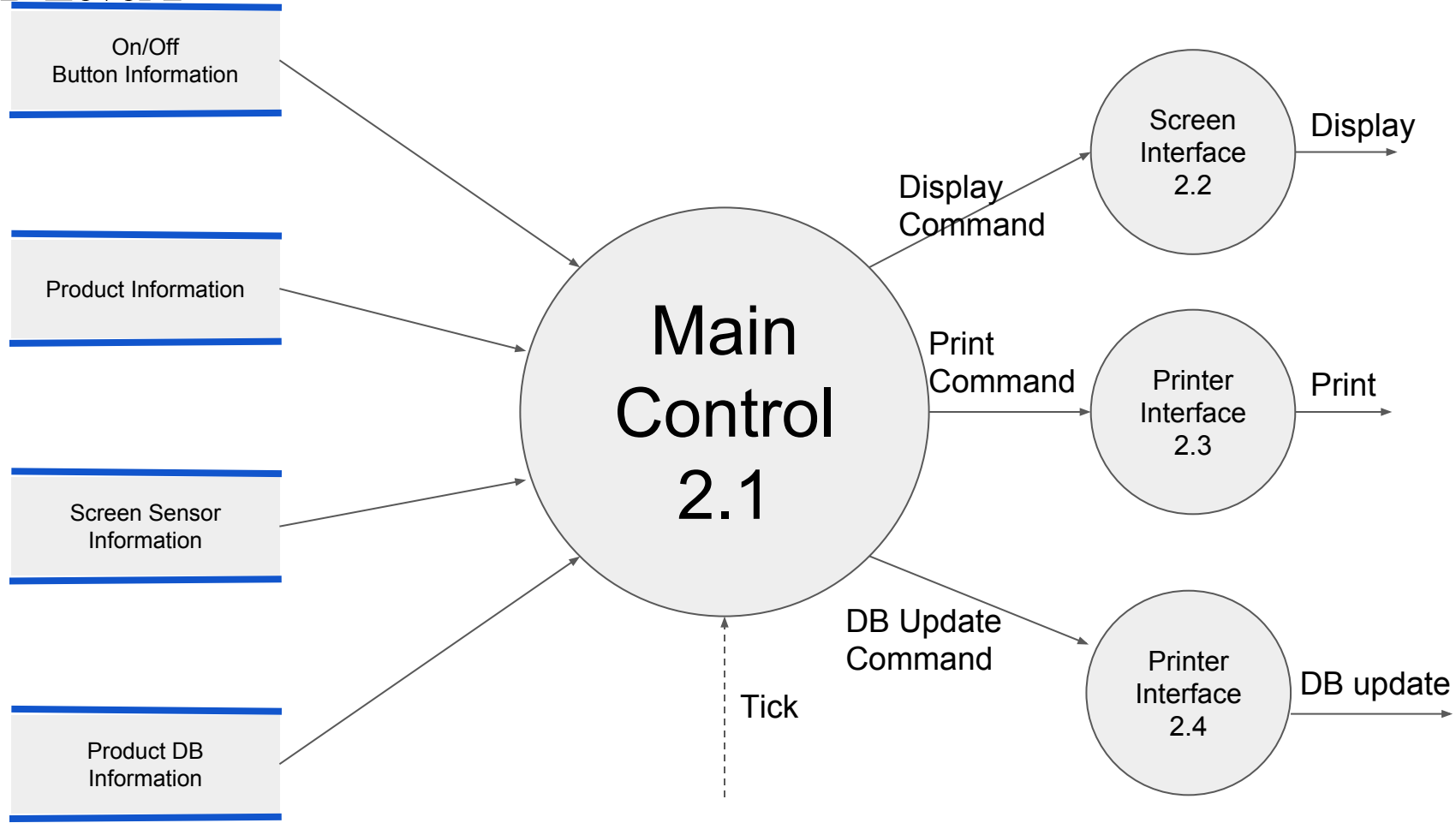
DFD Level 1



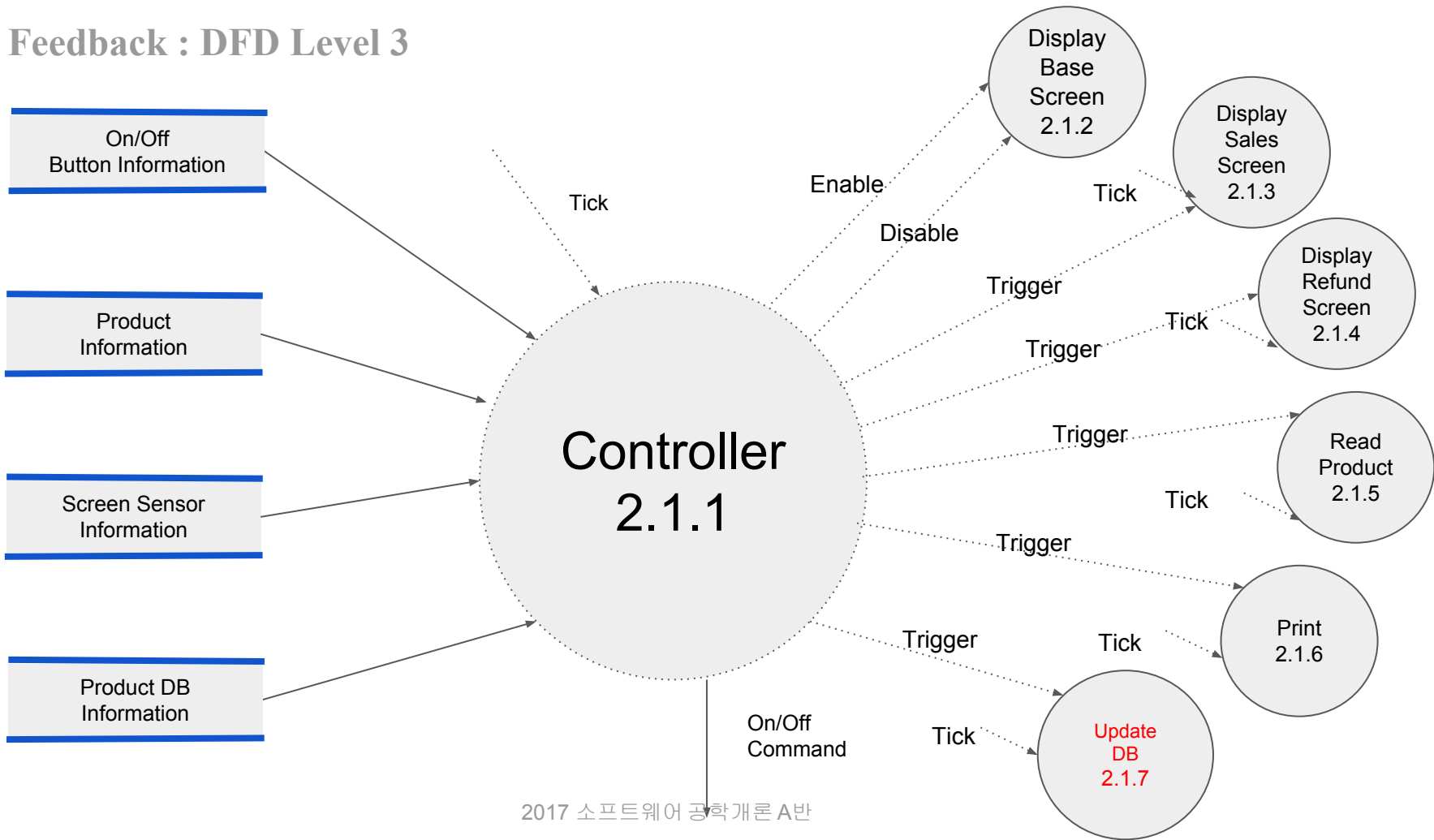
DFD Level 2



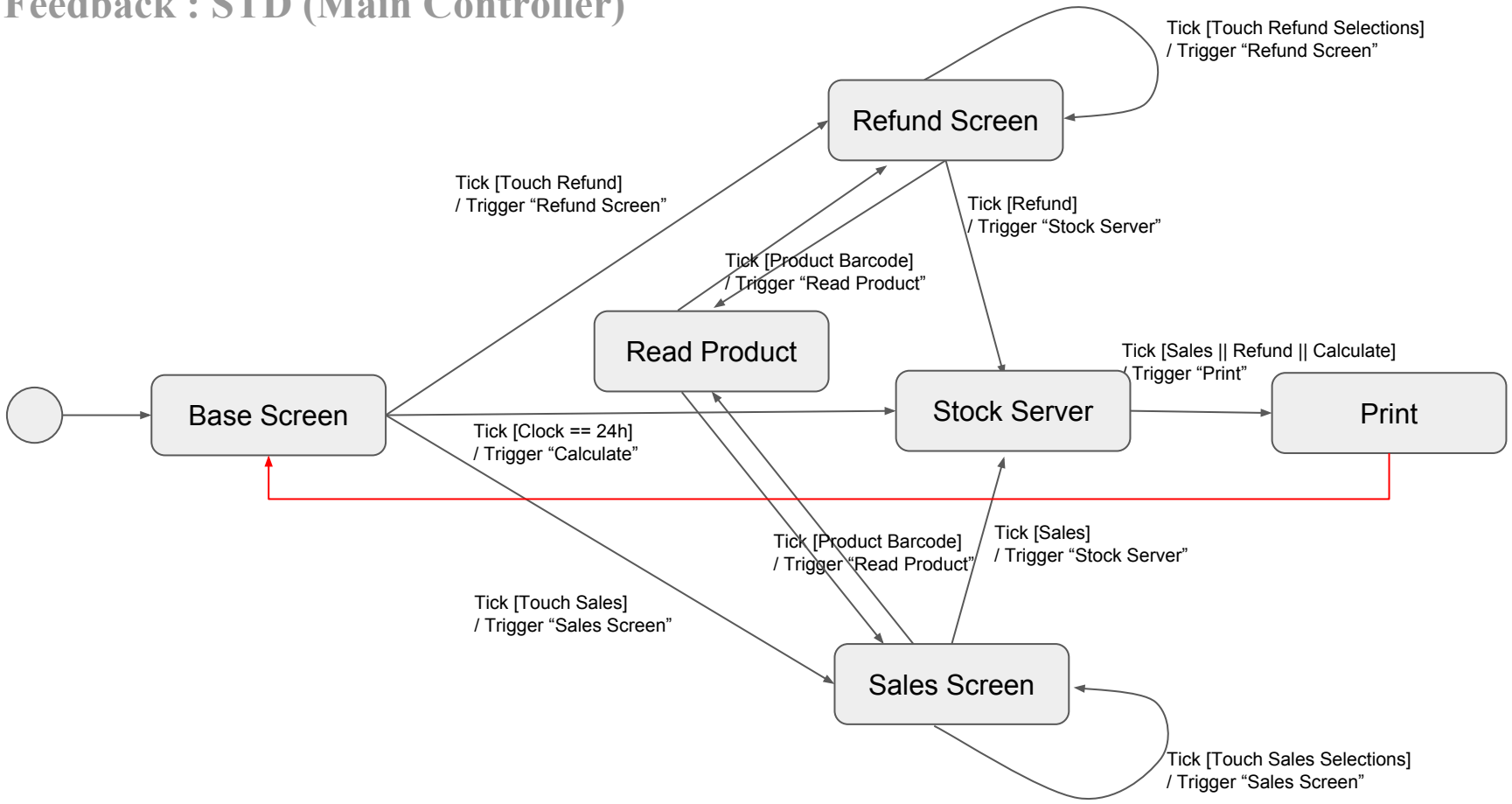
DFD Level 2



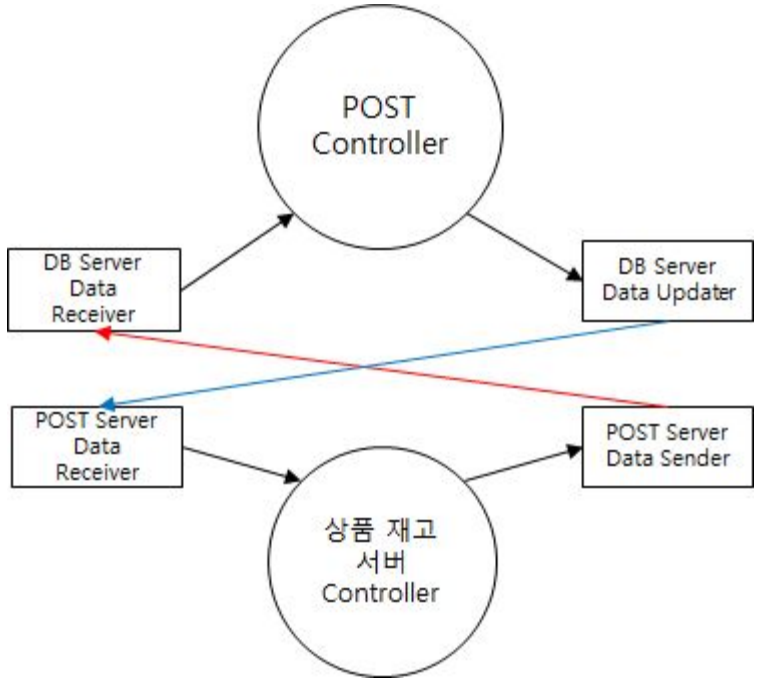
| SA Feedback : DFD Level 3



| SA Feedback : STD (Main Controller)

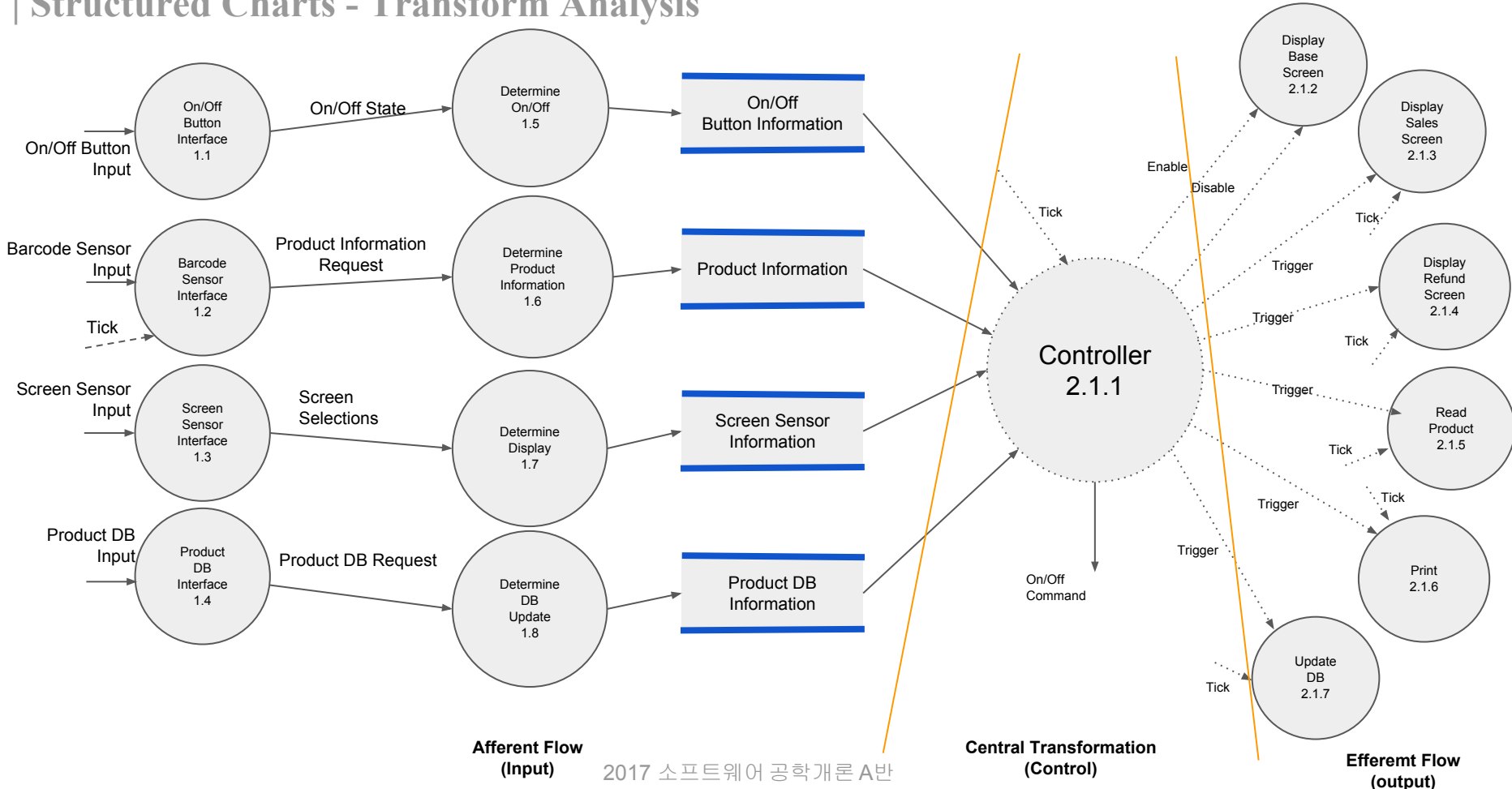


| SA Feedback : STD (Main Controller)



Structured Design

| Structured Charts - Transform Analysis

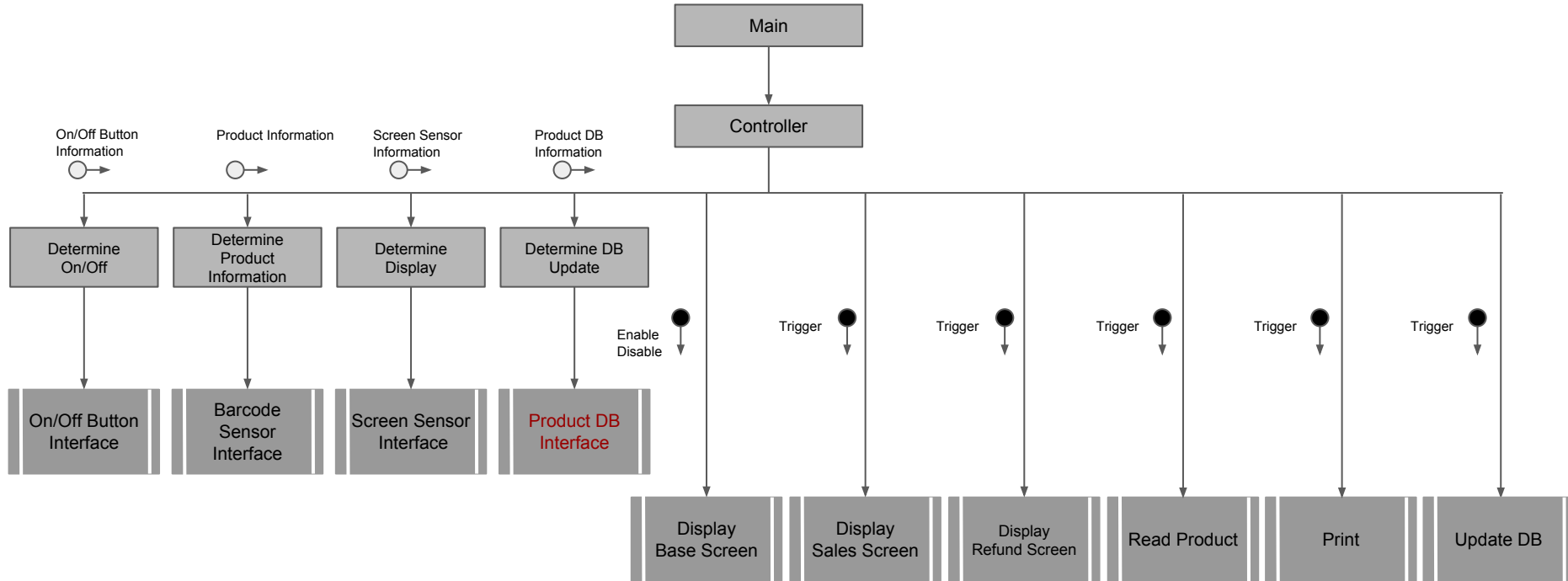


Afferent Flow (Input)

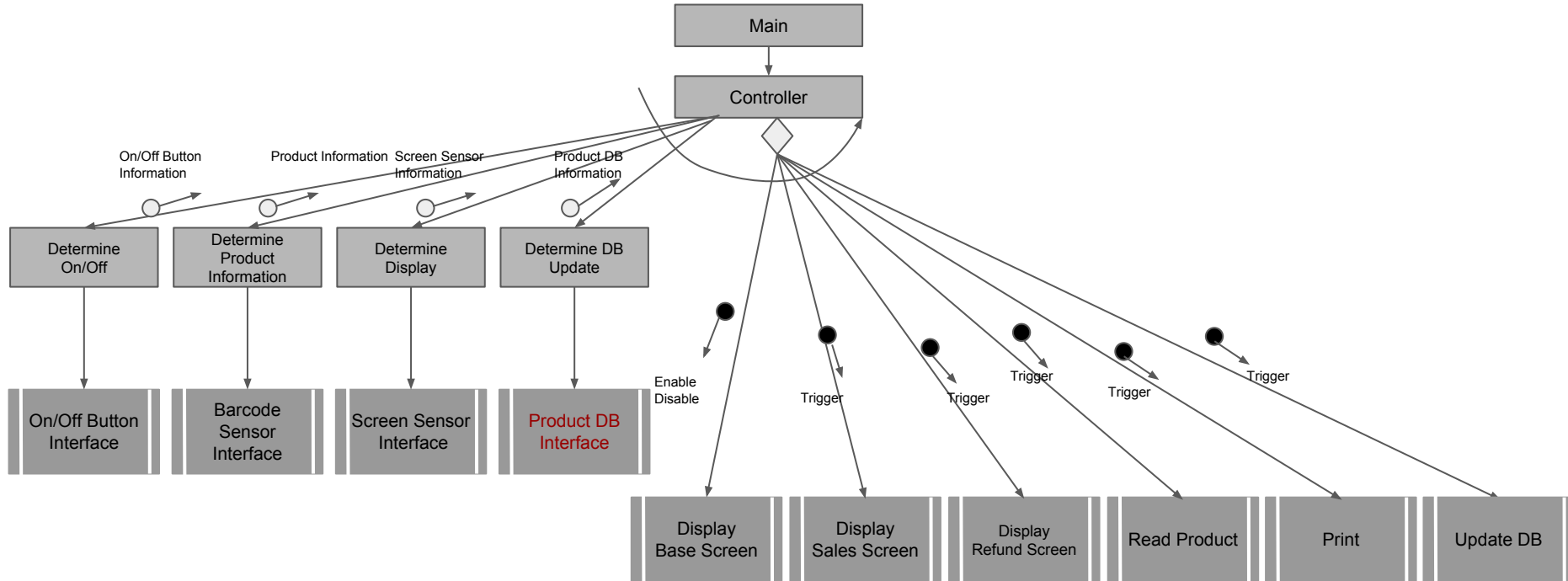
Central Transformation (Control)

Efferent Flow (output)

| Transform Analysis (Basic)



| Transform Analysis (Advanced)



- 코드 구현

안녕하세요 포스 시스템입니다.

보고계신 화면은 BaseScreen입니다. 해당 번호를 눌러주세요

- 1 -> SalesScreen화면으로 전환합니다.
- 2 -> RefundScreen화면으로 전환합니다.
- 3 -> 재고를 확인합니다.
- 4 -> 프로그램을 종료합니다.



1
SalesScreen입니다. 바코드를 찍어주세요 : 110
상품 수량을 입력해주세요 : 3
더 구매하실 상품이 있습니까? 있으면 0을 눌러주세요 : 없으면 1을 눌러주세요



1
drink가 3개이고 3600원입니다.
구매하신 상품의 총액은 3600입니다.종료 1, 진행 00
안녕하세요 포스 시스템입니다.

보고계신 화면은 BaseScreen입니다. 해당 번호를 눌러주세요

- 1 -> SalesScreen화면으로 전환합니다.
 - 2 -> RefundScreen화면으로 전환합니다.
 - 3 -> 재고를 확인합니다.
 - 4 -> 프로그램을 종료합니다.
- 3



3
=====
=====
상품명 : snack 상품 단가 : 1000 재고 수량 : 100
상품명 : icecream 상품 단가 : 1500 재고 수량 : 100
상품명 : fruit 상품 단가 : 3000 재고 수량 : 100
상품명 : water 상품 단가 : 500 재고 수량 : 100
상품명 : ramen 상품 단가 : 800 재고 수량 : 100
상품명 : drink 상품 단가 : 1200 재고 수량 : 97
상품명 : coffee 상품 단가 : 2000 재고 수량 : 100
=====
=====
종료 1, 진행 0
4



- 코드 구현

```
0
안녕하세요 포스 시스템입니다.
  보고계신 화면은 BaseScreen입니다. 해당 번호를 눌러주세요
1 -> SalesScreen화면으로 전환합니다.
2 -> RefundScreen화면으로 전환합니다.
3 -> 재고를 확인합니다.
4 -> 프로그램을 종료합니다.
2
RefundScreen입니다. 바코드를 찍어주세요 : 111
상품 수량을 입력해주세요 : 3
더 환불하실 상품이 있습니까? : yes :0 no :
1 : 1
coffee가 3개이고 6000원입니다.
환불하신 상품의 총액은 6000입니다.종료 1, 진행 00
안녕하세요 포스 시스템입니다.
  보고계신 화면은 BaseScreen입니다. 해당 번호를 눌러주세요
1 -> SalesScreen화면으로 전환합니다.
2 -> RefundScreen화면으로 전환합니다.
3 -> 재고를 확인합니다.
4 -> 프로그램을 종료합니다.
```



```
3
=====
=====
상품명 : snack 상품 단가 : 1000 재고 수량 :
100
상품명 : icecream 상품 단가 : 1500 재고 수량 :
100
상품명 : fruit 상품 단가 : 3000 재고 수량 :
100
상품명 : water 상품 단가 : 500 재고 수량 : 100
상품명 : ramen 상품 단가 : 800 재고 수량 : 100
상품명 : drink 상품 단가 : 1200 재고 수량 : 97
상품명 : coffee 상품 단가 : 2000 재고 수량 :
103
=====
=====
종료 1, 진행 00
안녕하세요 포스 시스템입니다.
  보고계신 화면은 BaseScreen입니다. 해당 번호를 눌러주세요
1 -> SalesScreen화면으로 전환합니다.
2 -> RefundScreen화면으로 전환합니다.
3 -> 재고를 확인합니다.
4 -> 프로그램을 종료합니다.
4
종료하겠습니다.종료 1, 진행 01
Program ended with exit code: 0
```

- 코드 미구현 요소

1. 영수증 출력
2. On_Off 버튼
3. 고객 화면 부분
4. 정산
5. Print 출력기능

- UNIT TEST Problem

1. 각각의 해당 사항에 대한 테스트 작성 방법 숙지에 대한 부족으로 테스트 프레임워크 적용에 실패.

2. 코드 직접 적용의 문제점 -> 각각의 return 값 적용으로 인해서, 테스트를 위해 코드를 수정해야 하는 문제가 발생

3. UTP의 수정과 UTR 작성

- UNIT TEST

```
main.c:39:20: error: function definition is not allowed here
void checkProduct(){
    ^
main.c:50:13: error: function definition is not allowed here
void Sales(){
    ^
main.c:85:14: error: function definition is not allowed here
void Refund(){
    ^
main.c:118:21: error: function definition is not allowed here
void BalanceCheck() {
    ^
main.c:128:12: error: function definition is not allowed here
int main() {
    ^
5 errors generated.
```

```
> test1...
> test2...
[FAIL] simple.c:24 : (test2) : mul()==21 fails

--- Results ---
Tests run:      2
Passes:         3
Failures:       1
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
```

- UNIT TEST

```
int main() {
    TestcheckProduct();
    TestSales();
    TestRefund();
    TestBalanceCheck();
}

void TestCheckProduct001() {
    int checkBarcode = 001;

    if (checkProduct(001) == 1000)
        printf("checkProduct 함수 성공");
    else
        printf("checkProduct 함수 실패");
}

void TestCheckProduct001() {
    int checkBarcode = 100;

    if (checkProduct(100) == 500)
        printf("checkProduct 함수 성공");
    else
        printf("checkProduct 함수 실패");
}

void TestCheckProduct001() {
    int checkBarcode = 111;

    if (checkProduct(111) == 2000)
        printf("checkProduct 함수 성공");
    else
        printf("checkProduct 함수 실패");
}
```