

# stm32f429-DMA Tutorial

제작자: 유영재

## 목차

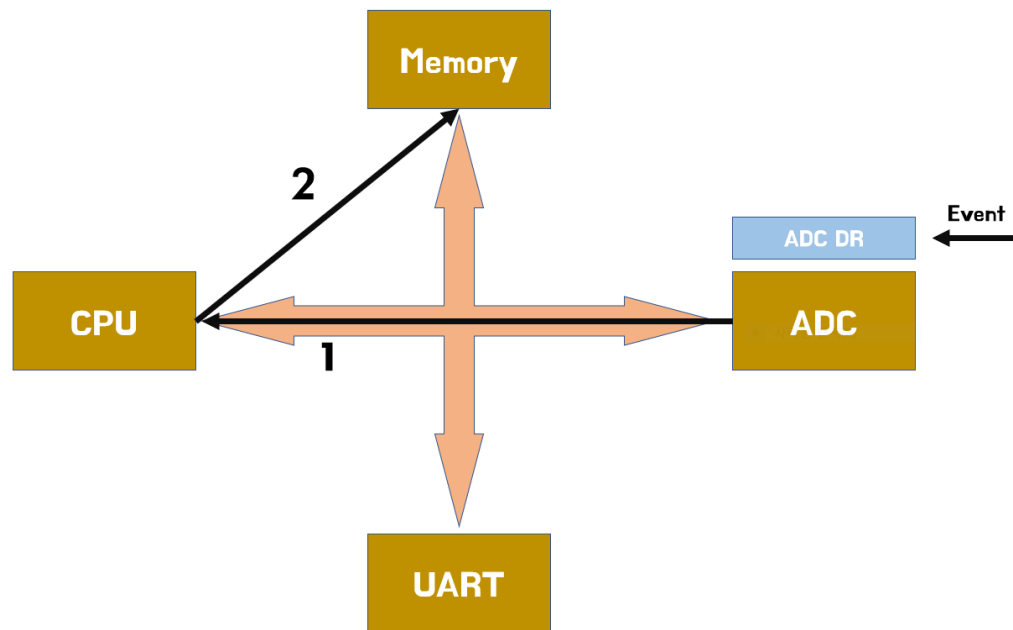
1.	DMA 소개 .....	3
1.1	Master-slave system.....	3
1.2	DMA 사용 예 .....	4
1.3	MCU Block Diagram .....	5

# 1. DMA 소개

- DMA 는 Direct Memory Access 의 약자로 최근 MCU 에는 기본적으로 들어가 있는 Peripheral 중 하나다. 현재 우리가 다룰 arm-cortex M4 기반의 stm32f429 도 총 2 개의 dma 컨트롤러가 내장되어 있다. dma 를 이해하기 위해선 arm-cortex 의 버스 시스템을 이해하고 있어야 한다

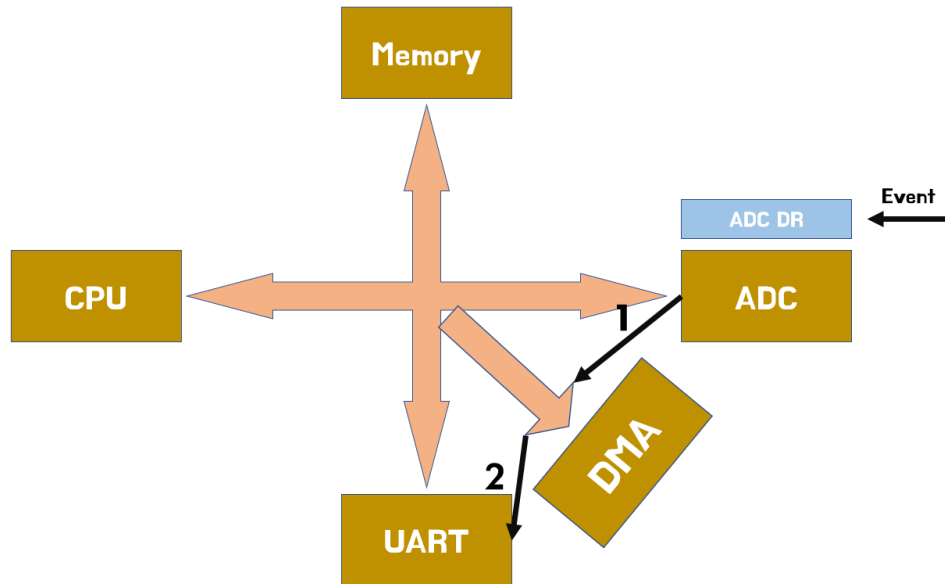
## 1.1 Master-slave system

아래 그림은 ARM의 버스 시스템을 나타낸 것이다. ARM 버스 시스템은 기본적으로 CPU와 주변 Peripheral들은 버스로 연결되어 있는 것을 볼 수 있다



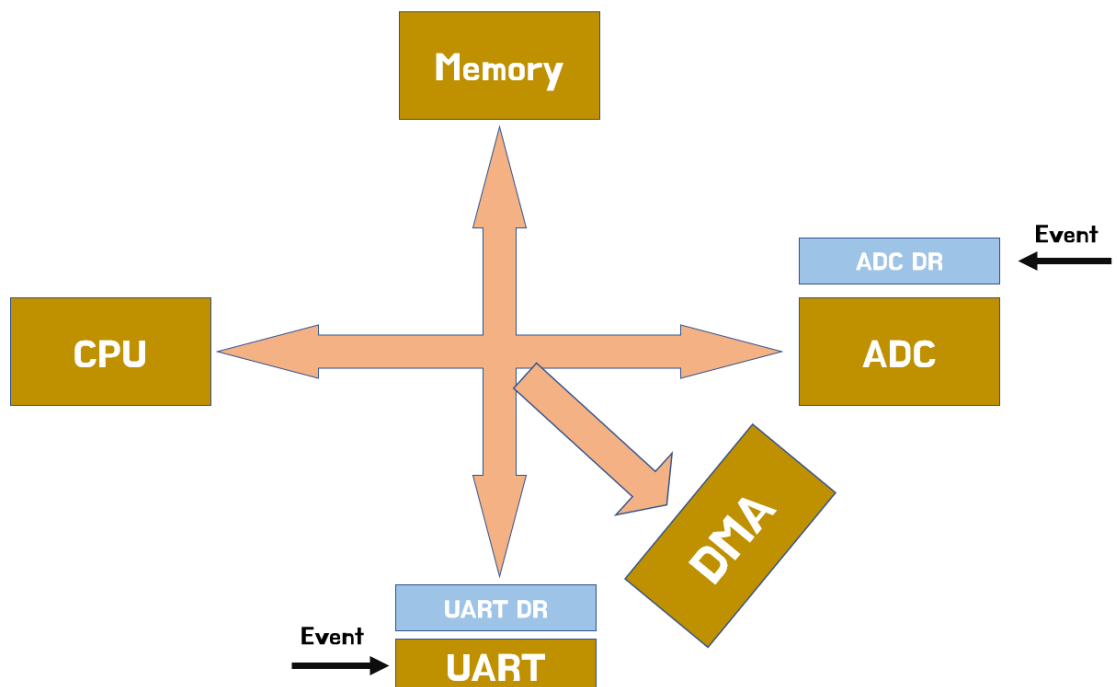
- 하나의 상황을 예를 들어서 설명을 해보려고 한다. ADC에 이벤트가 발생해서 digitize 된 값이 ADC 데이터 레지스터로 쓰여지게 된다
- 그러면 ADC는 CPU에게 Event를 발생시켜서 CPU는 해당 데이터를 프로세서 레지스터에 저장한 이후 SRAM 메모리에 쓸 수 있게 된다
- 위에서 설명했듯이 총 2개의 경로로 데이터는 메모리에 쓰여지게 된다. 그러면 바로 ADC DR 값이 메모리로 쓸 수는 없을까?
- 답은 그렇게 할 수 없다. 왜냐하면 명령어 중 하나인 load는 CPU만 가지고 있기 때문이다. 즉 각 주변장치들은 두뇌가 없기 때문에 해당 명령을 실행할 수 없다. 그래서 매번 CPU의 도움을 받기 위해서 Event를 발생시키는 것이다
- 그래서 버스 시스템에서 버스를 컨트롤 할 수 있는 기능을 가지고 있는 것은 master라고 부르며, 아닌 것은 slave라고 지칭한다. 그래서 slave끼리는 버스를 끼고 데이터 공유가 불가능한 것이다
- 그러면 CPU 말고도 master 역할을 대신 할 수 있는 것은 없나? 그 역할을 하는 것이 바로 DMA controller다.

- G. DMA 컨트롤러가 master라고 지칭할 수 있어도, 명령어(instruction)를 제공하는 장치는 CPU밖에 존재하지 않는다. 다만 DMA가 하드웨어 로직으로 CPU의 일을 offload할 수 있게 설계되어 있을 뿐이다. 그래서 도움을 받아 버스를 control 할 수 있게 된다



## 1.2 DMA 사용 예

- DMA가 CPU load를 덜어주는 역할을 한다고 하지만, 언제 사용하게 되는가? 단순히 CPU가 하나의 일만 수행을 한다면 굳이 사용할 필요는 없다. 다음과 같은 예를 들어보기로 한다



- A. 두 인터럽트가 동시에 config되어 있고, ADC, UART 모두 동시에 들어오는 상황이라고 가정  
을 한다. 인터럽트로 받은 데이터를 모두 메모리에 저장해야 하는 상황이다
  - B. ADC 우선순위가 더 높은 상황이고, 데이터는 두 주변장치에서 지속적으로 들어오고 있다. 그  
래서 ADC는 데이터가 지속적으로 SRAM에 저장되지만, UART는 그렇지 않다
  - C. 이럴 때 UART 데이터 손실을 막기 위해서 UART 데이터는 DMA 컨트롤러로 처리를 하게  
하는 것이다. 따라서 DMA를 거쳐 SRAM으로 저장되게 된다
- 그리고 두번째 이유로 뒤에 실험을 하겠지만, **ARM ON일 때와 ARM OFF/DMA ON일 때의 전류  
를 비교하면 차이가 생기게 된다. 전력을 아끼는 쪽이 DMA이기 때문에 좋은 Application에는  
DMA 사용이 반드시 따라오게 된다**

### 1.3 MCU Block Diagram

MCU Block Diagram은 DMA를 이해하는 데 반드시 숙지해야 하는 부분이다. RM 문서 2장  
Memory and bus architecture을 보면 위에서 언급한 master/slave 개념이 드러나고 있다.  
stm32f429에는 2개의 DMA를 포함해서 10개의 master 버스가 존재한다

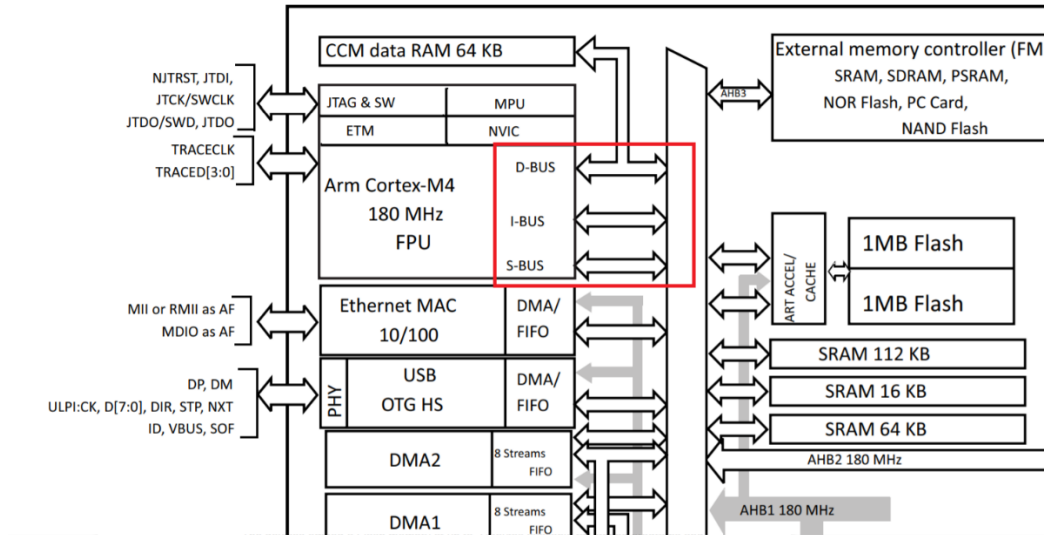
In the STM32F42xx and STM32F43xx devices, the main system consists of 32-bit multilayer  
AHB bus matrix that interconnects:

- Ten masters:
  - Cortex<sup>®</sup>-M4 with FPU core I-bus, D-bus and S-bus
  - DMA1 memory bus
  - DMA2 memory bus
  - DMA2 peripheral bus
  - Ethernet DMA bus
  - USB OTG HS DMA bus
  - LCD Controller DMA-bus
  - DMA2D (Chrom-Art Accelerator<sup>™</sup>) memory bus
- Eight slaves:
  - Internal Flash memory ICode bus
  - Internal Flash memory DCode bus
  - Main internal SRAM1 (112 KB)
  - Auxiliary internal SRAM2 (16 KB)
  - Auxiliary internal SRAM3 (64 KB)
  - AHB1peripherals including AHB to APB bridges and APB peripherals
  - AHB2 peripherals
  - FMC

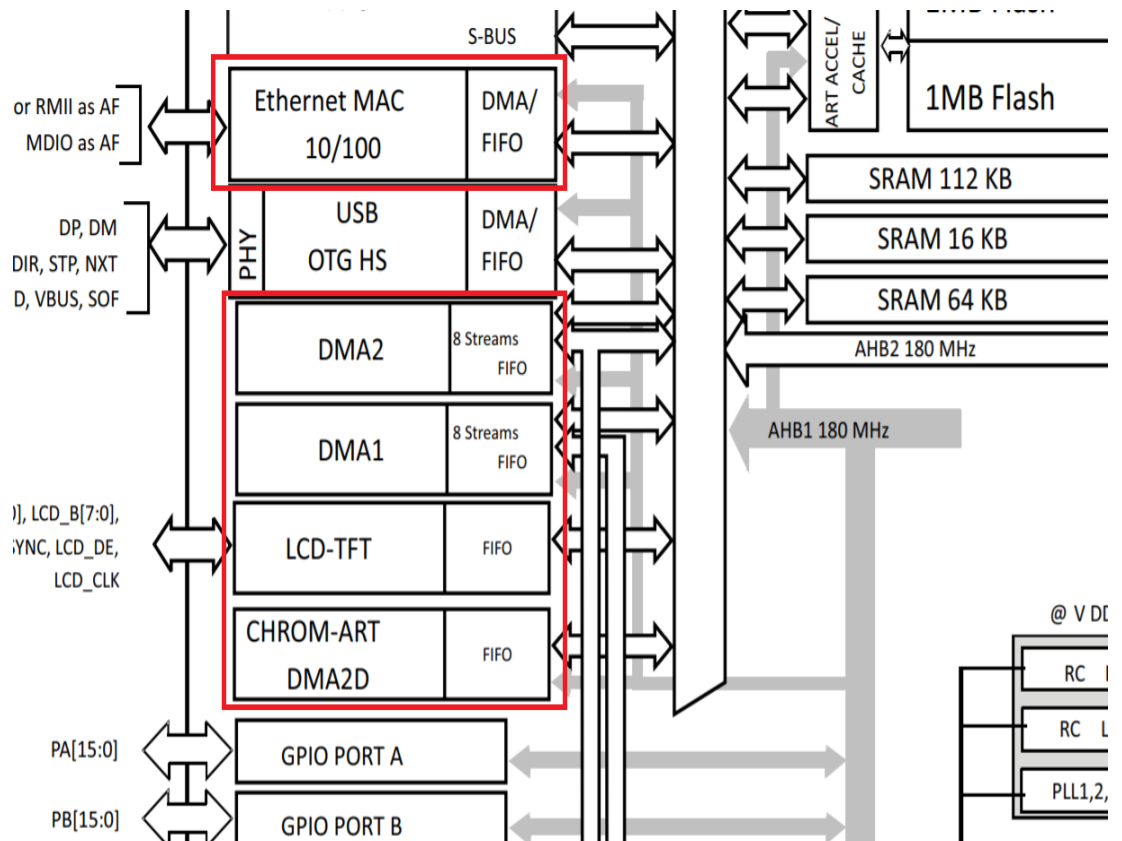
The bus matrix provides access from a master to a slave, enabling concurrent access and  
efficient operation even when several high-speed peripherals work simultaneously. The 64-  
Kbyte CCM (core coupled memory) data RAM is not part of the bus matrix and can be  
accessed only through the CPU. This architecture is shown in [Figure 2](#).

## A. Master

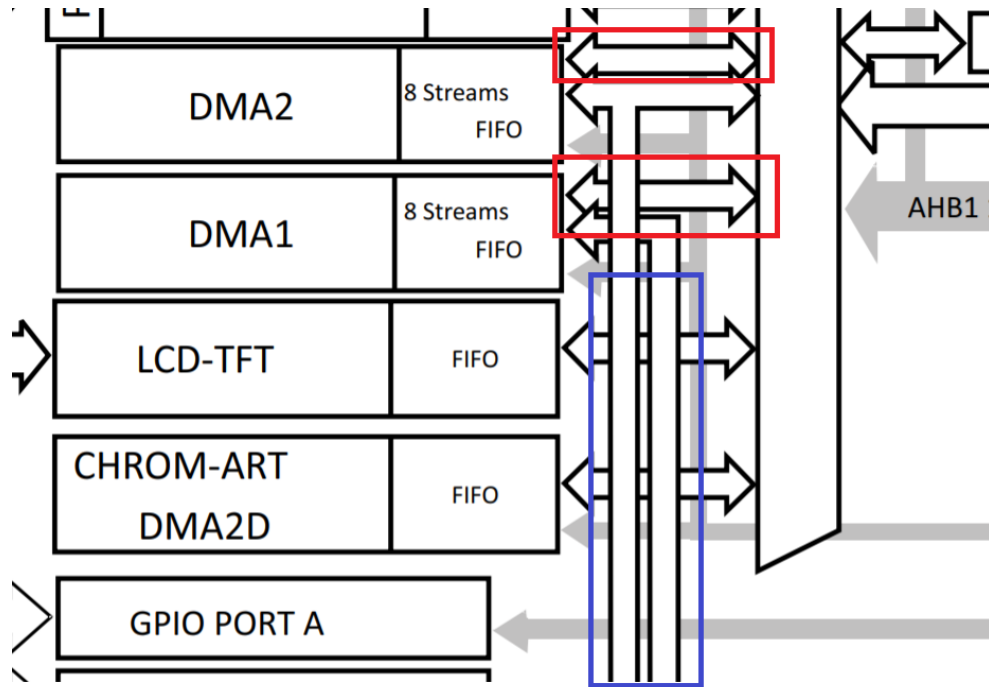
### a. Core I-bus, S-bus, D-bus



### b. 다른 Master



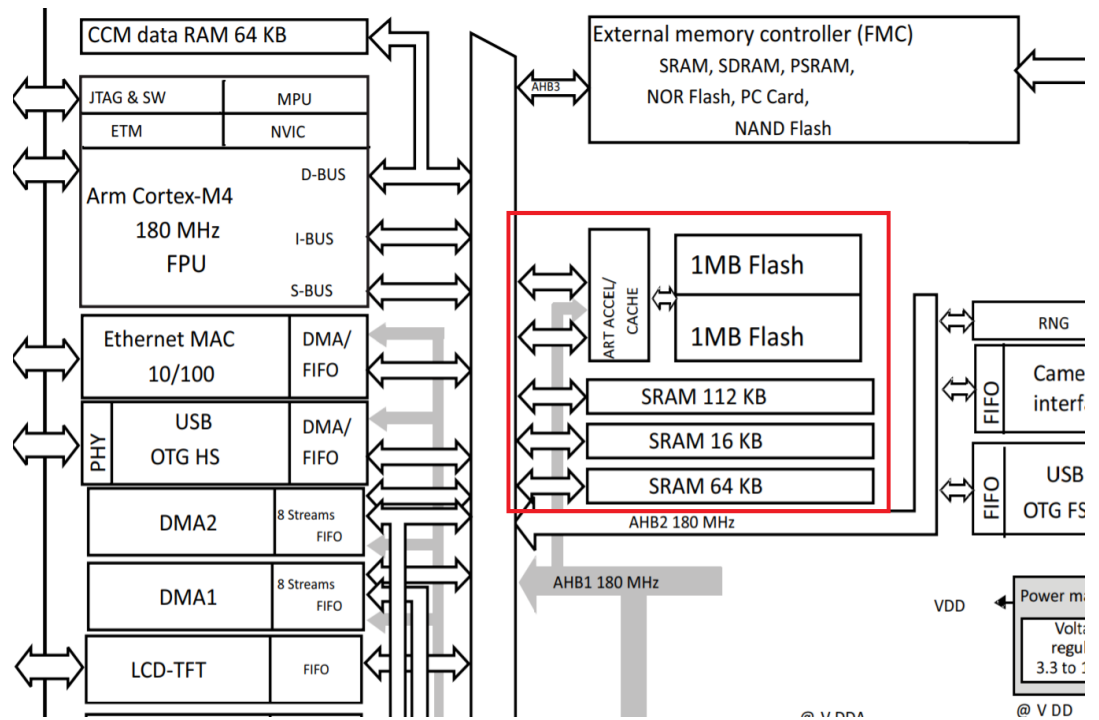
참고로 DMA controller에서 두 갈래의 길로 나뉘진다. 하나는 memory쪽 버스와 나머지는 peripheral쪽 버스가 된다. 아래 버스에서 빨간색이 memory, 파란색이 peripheral로 해당된다



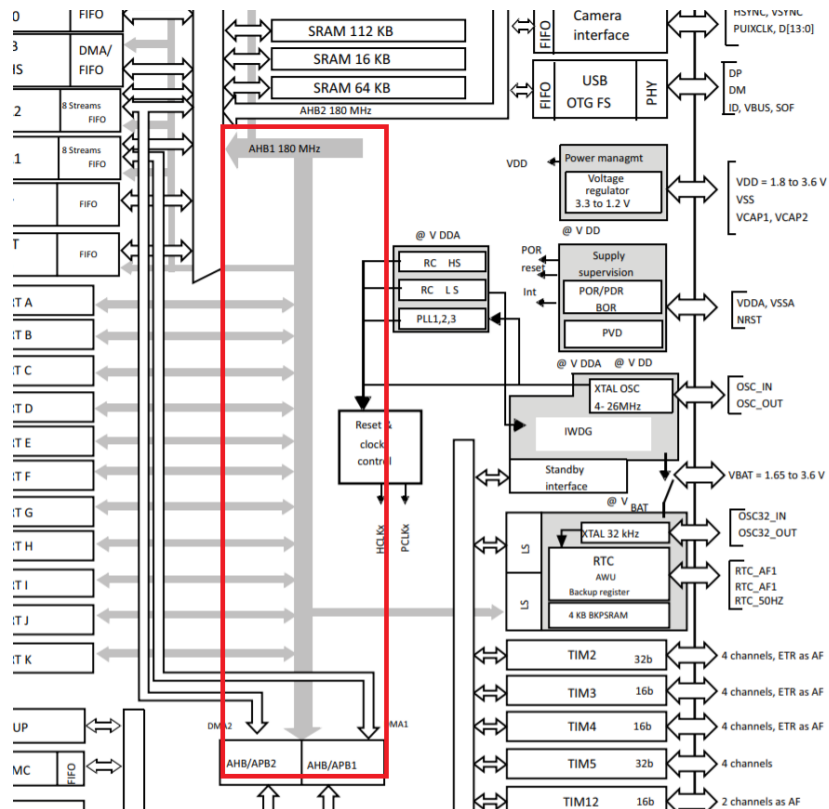
따라서 DMA 컨트롤러를 통해서 M2M, P2M, M2P가 가능하게 된다

## B. Slave

### a. Flash I-code, D-code, SRAM



### b. AHB1은 APB1, 2 버스를 포함하고 있다





The diagram illustrates the system architecture of the Pico-32 microcontroller. On the left, a vertical bus is shown with multiple bidirectional arrows indicating data flow. The main system bus is the AHB2 180 MHz, which connects to several components: an ART ACCEL/CACHE, two 1MB Flash blocks, and three SRAM blocks (112 KB, 16 KB, and 64 KB). A secondary AHB1 180 MHz bus is also shown. The AHB2 180 MHz bus connects to a red-outlined block containing the RNG, Camera interface (with a FIFO), and USB OTG FS (with a FIFO and PHY). This block interfaces with external signals: HSYNC, VSYNC, PUIXCLK, D[13:0], DP, DM, ID, VBUS, and SOF. Below the main bus, a Power managmt block contains a Voltage regulator (2.3 to 1.2 V) and is connected to VDD and VSS. The VDD supply is specified as 1.8 to 3.6 V.

[illegible]