

arm

SysTick Timer

Learning Objectives

- By the end of this module, you will be able to:
 - Introduce the System Timer (SysTick) and associated functionality
 - Program SysTick for a specific time interval and trigger interrupt
 - Measure the cycle count of a function using SysTick

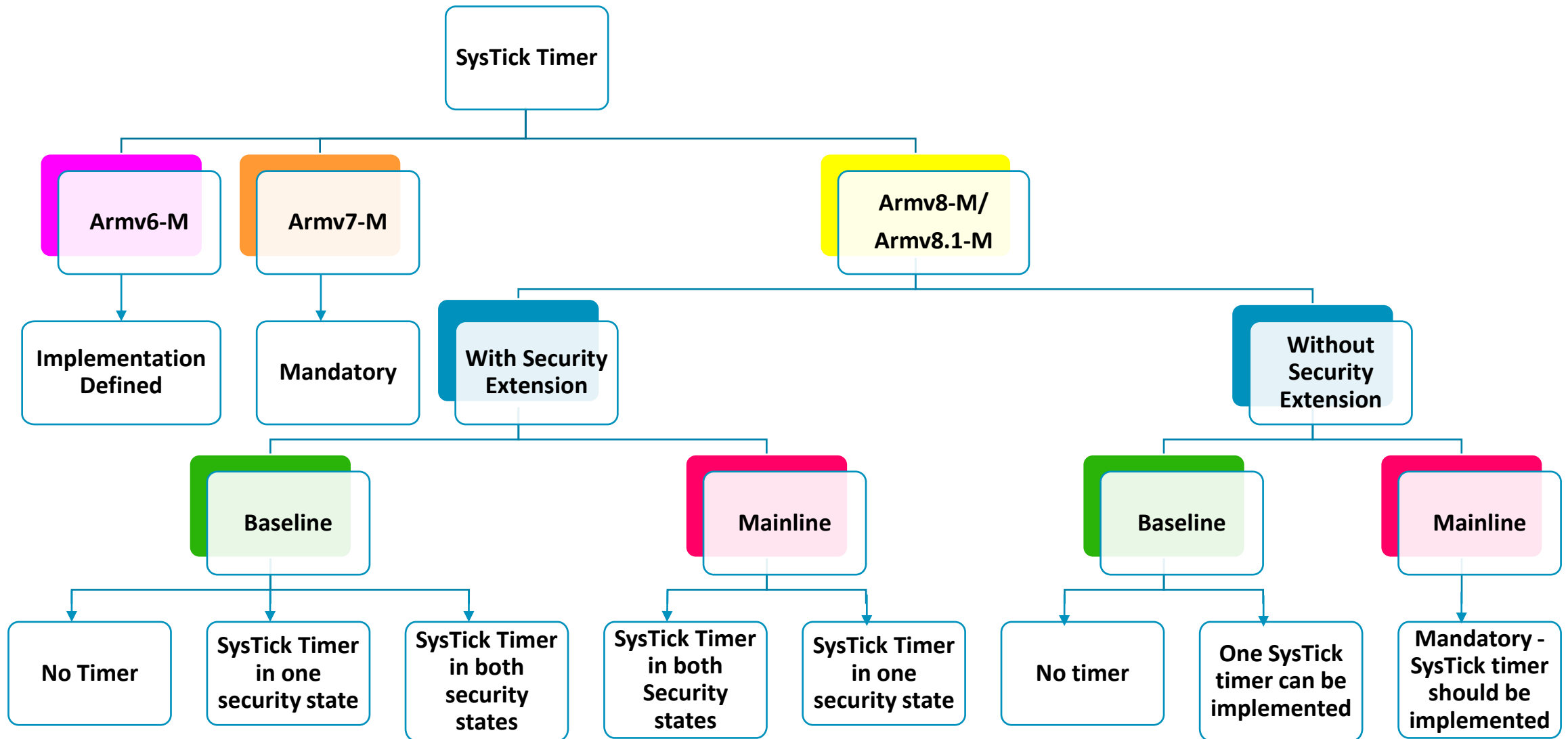
Agenda

- **Architecture and Software Usage**
 - **Systick Timer (SysTick) Overview**
 - Using CMSIS for SysTick

The SysTick Timer

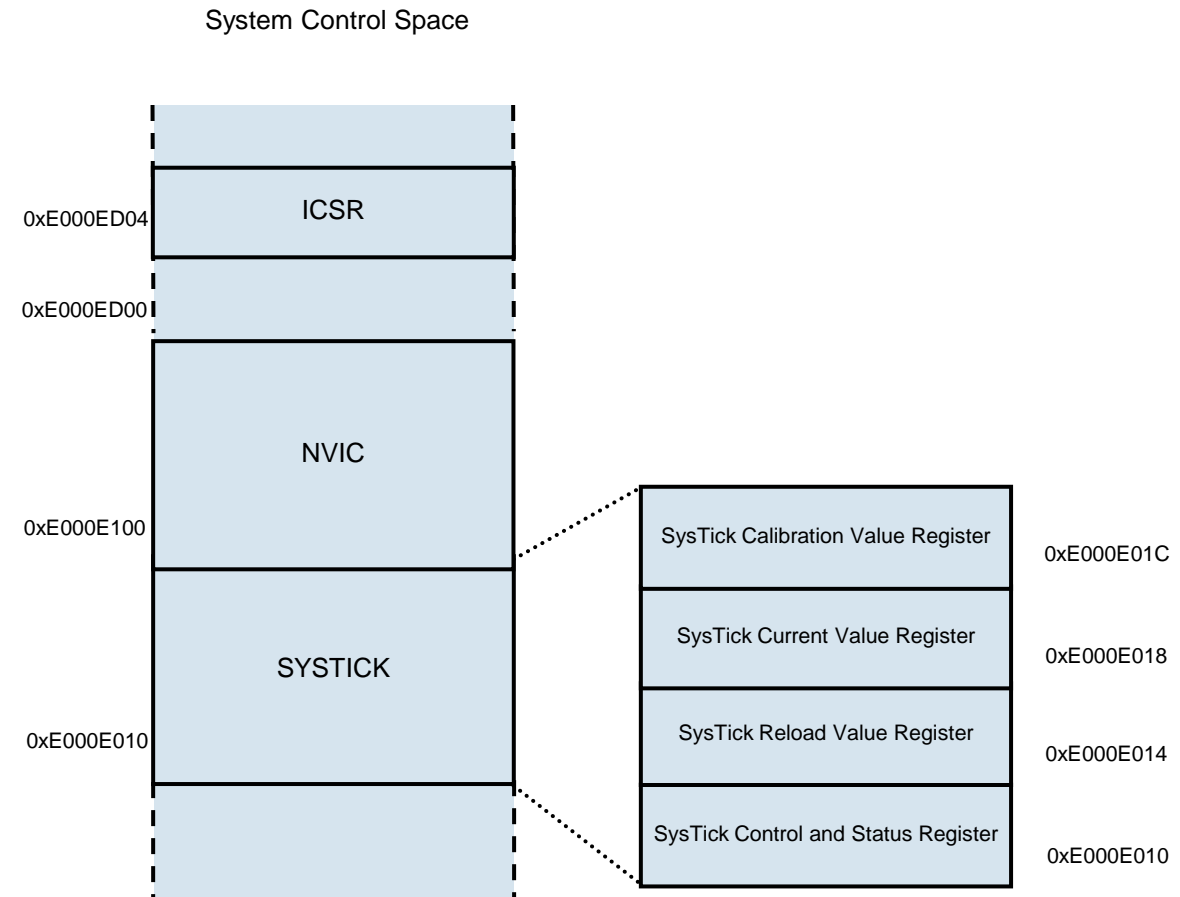
- SysTick is a 24-bit counter
 - Simple decrementing counter
 - Reload-on-zero
 - Clear-on-write
- Flexible control mechanism using four system registers
- Provides internal system timer (SysTick)
 - SysTick Exception number is 15
 - SysTick Vector offset address is 0x3C
- SysTick can be used in different ways
 - RTOS tick timer
 - Dynamic clock management
- Inclusion of the timer depends on the architecture variant of the implementation
 - When SysTick is not implemented in Armv6-M / Armv8-M – Baseline the vector is reserved

SysTick Implementation Options

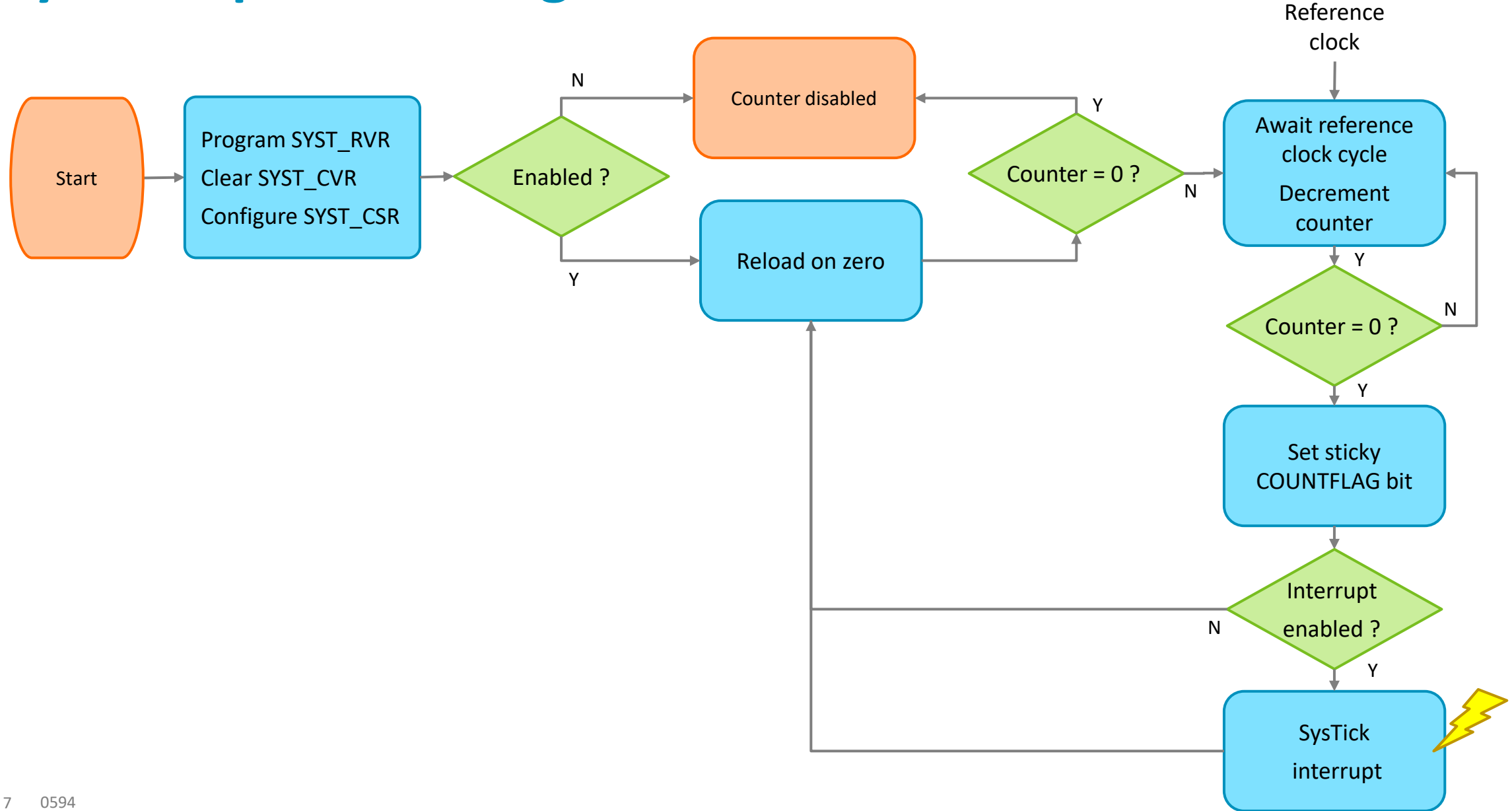


SysTick Registers

- SysTick Control and Status Register (SYST_CSR)
 - Select the timing source
 - Enable the counter
 - Enable the SysTick interrupt
 - Determine counter status
- SysTick Reload Value Register (SYST_RVR)
 - Counter reload value
- SysTick Current Value Register (SYST_CVR)
- SysTick Calibration Value Register (SYST_CALIB)
 - Describes the timing reference available on this chip



SysTick operation diagram



SysTick Operation

- Program the Reload Value Register with the desired reload value
 - Zero disables SysTick
 - Reload value = number of cycles – 1
- Clear the Current Value Register to trigger an immediate reload
 - This register value is UNKNOWN out of reset. Hence software need to clear this register before counter starts counting
- Enable the counter using Status & Control Register
 - Enable SysTick interrupt if desired
- Current value is decremented each reference clock cycle
- Current value is auto-reloaded after a transition from 1 to 0
 - Counter wrapping flag is set
 - SysTick interrupt is generated if interrupt generation is enabled

Agenda

- **Architecture and Software Usage**
 - SysTick Timer (SysTick) Overview
 - **Using CMSIS for SysTick**

CMSIS-Core: SysTick Configuration

SysTick_type

Structure type to access System Timer registers

CTRL : SysTick Control and Status registers (0xE000E010)

LOAD : SysTick Reload Value register (0xE000E014)

VAL : SysTick Current Value register (0xE000E018)

CALIB : SysTick Calibration register (0xE000E01C)

uint32_t SysTick_Config (uint32_t ticks)

CMSIS-Core function for SysTick configuration

Initializes the system timer and interrupt

Counter in free running mode

__Vendor_SysTickConfig

When defined to 1, the standard function SysTick_Config is excluded

In this case, the Device Header File <device.h> must contain a vendor specific implementation of this function

```
__STATIC_INLINE uint32_t SysTick_Config
(uint32_t ticks)
{
    if ((ticks - 1UL) > SysTick_LOAD_RELOAD_Msk)
    {
        /*Reload value impossible */
        return (1UL);
    }
    /*set reload register*/
    SysTick->LOAD = (uint32_t) (ticks - 1UL);

    /* set Priority for SysTick Interrupt */
    NVIC_SetPriority (SysTick_IRQn,
                     (1UL << __NVIC_PRIO_BITS) - 1UL);

    /* Load the SysTick Counter Value */
    SysTick->VAL = 0UL;

    /* Enable SysTick IRQ and SysTick Timer */
    /* Function successful */
    SysTick->CTRL = SysTick_CTRL_CLKSOURCE_Msk |
                   SysTick_CTRL_TICKINT_Msk |
                   SysTick_CTRL_ENABLE_Msk;

    return (0UL);
}
```

Measure cycle count via SysTick registers



```
uint32_t SysTick_start_timer()
{
    /* Set Reload Register to maximum */
    SysTick->LOAD = SysTick_LOAD_RELOAD_Msk ;

    /* Clear Current Value Register by
       writing any value */
    SysTick->VAL = 0x1;

    /* Enable SysTick Timer */
    SysTick->CTRL = SysTick_CTRL_CLKSOURCE_Msk |
                   SysTick_CTRL_TICKINT_Msk |
                   SysTick_CTRL_ENABLE_Msk;

    return (0);
}
```

```
/* SysTick handler overflow variable */
extern uint32_t SysTick_overflow_count;

uint32_t SysTick_stop_timer()
{
    uint32_t count;
    /* Clear enable to stop timer */
    SysTick->CTRL = ~(SysTick_CTRL_CLKSOURCE_Msk |
                     SysTick_CTRL_TICKINT_Msk |
                     SysTick_CTRL_ENABLE_Msk);

    /* Read Current Value Register and concatenate
       with SysTick timer overflow count */
    count = (SysTick_overflow_count << 24) |
            (SysTick->VAL);

    /* Reset counter */
    SysTick->VAL = 0x1;
    return count;
}
```

arm

Thank You

Danke

Gracias

Grazie

谢谢

ありがとう

Asante

Merci

감사합니다

धन्यवाद

Kiitos

شكراً

ধন্যবাদ

תודה