

Cortex-M Microcontroller မိတ်ဆက်

မိုက်ခရိုကွန်ထရိုလာ ဒီဇိုင်းတစ်ခုမှာ ပရိုဆက်ဟာ subsystem တစ်ခုသာဖြစ်ပြီး အခြားသော subsystem များဖြစ်ကြတဲ့ memory (volatile and non-volatile) , system bus တွေ digital I/O နဲ့ analog I/O တွေ၊ I2C တို့, SPI တို့, UART တို့ကို communication peripheral တွေ clock နဲ့ power management အပိုင်း တွေ စတဲ့ subsystem ပေါင်းများစွာနဲ့ ပေါင်းစပ်ဖွဲ့စည်းတည်ဆောက်ထားပါတယ်။ ARM Cortex-M ကို အဓိက CPU အနေနဲ့သုံးတဲ့ microcontroller ထုတ်လုပ်တဲ့ ကုမ္ပဏီအများစုမှာ CPU မှအပ အခြားသော System ရဲ့တည်ဆောက်ပုံတွေက မတူညီနိုင်ပါဘူး။

ARM architecture ဟာ အသုံးများတဲ့ architecture တွေထဲကတစ်ခုဖြစ်ပါတယ်။ အစောပိုင်းထွက်ခဲ့တဲ့ ARM architecture တစ်ခုကတော့ ARMv4/v4T ဖြစ်ပြီး ARM7TDMI ပရိုဆက်ဆာ တွေမှာ အဲဒီ ARMv4/v4T တည်ဆောက်ပုံရှိပါတယ်။ အသုံးများတဲ့ ပရိုဆက်ဆာတွေက တစ်ခုဖြစ်ခဲ့ပြီး Nokia6110 ဆဲလ်ဖုန်းမှာ အဲဒီ ARM7TDMI ပရိုဆက်ဆာကို အခြေခံတည်ဆောက်ထားပါတယ်။ ARMv5/5E architecture ကတော့ ARM9/9E family ပရိုဆက်ဆာတွေနဲ့အတူ မိတ်ဆက်လာပါတယ်။ ၎င်းမိတ်ဆက်လာတဲ့ ARMv5/5E architecture မှာ တော့ DSP (digital signal processing) ကိုထပ်ပေါင်းဖြည့်စွက်လာပါတယ်။ ၎င်းနောက် ARM11 family processor တွေနဲ့အတူ ARMv6 architecture ပေါ်ပေါက်လာပြီး Single-instruction multiple-data (SIMD) ရယ် Memory နဲ့ပတ်သတ်တဲ့ feature တွေရယ်ထပ်ပေါင်းဖြည့်စွက်လာပါတယ်။

ARM Architecture အပေါ်အခြေခံတည်ဆောက်ထားတဲ့ ပရိုဆက်ဆာတွေမှာ အုပ်စုသုံးခုရှိပါတယ်။

- Cortex-M :

မိုက်ခရိုကွန်ထရိုလာ အခြေခံတဲ့ Embedded Systems များအတွက် ဒီဇိုင်းပြုလုပ်ထားတာကတော့ Cortex-M microcontroller ဖြစ်ပါတယ်။

- Cortex-A :

ဆဲလ်ဖုန်းဈေးကွက်ကို အဓိကထားပြီးတော့ high performance(စွမ်းဆောင်ရည်မြင့်) application တွေ အတွက် ဒီဇိုင်းပြုလုပ်ထားတာဖြစ်ပါတယ်။

- Cortex-R :

Cortex-R ကတော့ real-time application တွေအတွက်ဖြစ်ပါတယ်။

ARM Cortex-M ကိုအခြေခံထားတဲ့ microcontroller တွေမှာ building block ၅ ခုရှိပါတယ်။

1. Microcontroller Core
2. Nested Vectored interrupt controller
3. Bus system and bus matrix

4. Memory and peripherals

5. Debug system

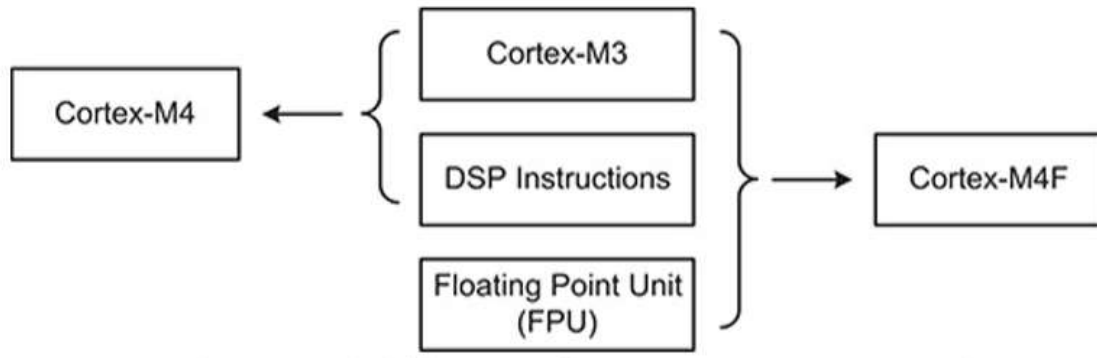
ARM processor architecture အမျိုးနဲ့ သူတို့ရဲ့ ပြောင်းလဲသွားပုံ အဆင့်ဆင့်ကိုအောက်တွင်ပြထားပါသည်။

Processor Family	ARM7TDMI	ARM9E	ARM11, Cortex-M0, Cortex-M1	Cortex-M, Cortex-R, Cortex-A
Processor Cores	ARM7TDMI, 920T, Intel StrongARM	ARM926, 946, 966, Intel XScale	ARM1136, 1176, 1156T2 Cortex-M0, Cortex-M1 (FPGA)	Cortex-M3/M4, Cortex-R4, Cortex-A8
Architecture Version	ARMv4/v4T	ARMv5/v5E	ARMv6	ARMv7

ပုံ() ARM processor architecture အမျိုးနဲ့ သူတို့ရဲ့ ပြောင်းလဲသွားပုံ

Microprocessor Architecture

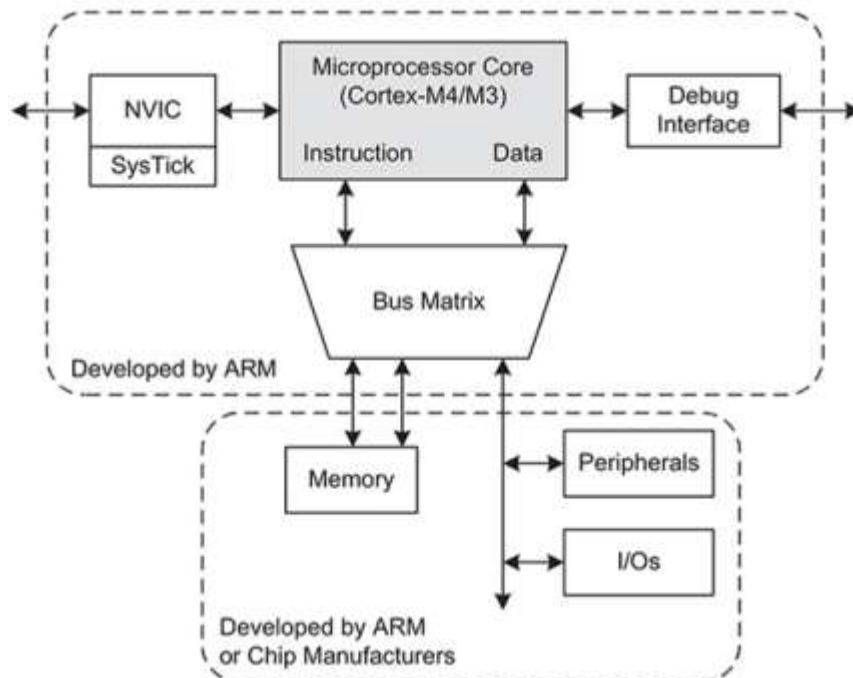
Cortex-M microcontroller တွေဟာ 32-bit RISC processor architecture အပေါ်မှာအခြေခံတည်ဆောက်ထားပါတယ်။ လက်ရှိထွက်နေတဲ့ Cortex-M ပရိုဆက်ဆာတွေမှာ Cortex-M0, Cortex-M1, Cortex-M3 နဲ့ Cortex-M4 family တွေရှိပါတယ်။ Cortex-M0 နဲ့ Cortex-M1 တွေဟာ ARMv6 architecture ကိုအခြေခံတည်ဆောက်ထားတာဖြစ်ပြီးတော့ Cortex-M3 နဲ့ Cortex-M4 တို့ဟာ ARMv7 architecture အပေါ်မှာ အခြေခံတည်ဆောက်ထားတာဖြစ်ပါတယ်။ Cortex-M3 နဲ့ Cortex-M4 တို့ရဲ့ အဓိက ခြားနားချက်ကတော့ DSP instruction ပါဝင်ခြင်းဖြစ်ပါတယ်။ Cortex-M4F ဆိုတဲ့နေရာမှာ 'F' ဆိုတာက Floating Point Unit(FPU) ပါဝင်ခြင်းဖြစ်ပါတယ်။ Cortex-M4 ပရိုဆက်ဆာတွေမှာတော့ FPU မပါဝင်ပါဘူး။ M4F တွေမှာဘဲ FPU ပါဝင်ပါတယ်။ ကျွန်တော်လက်ရှိသုံးနေတဲ့ Texas Instrument က Tivc C Launchpad (TM4C123G) ဟာ Cortex-M4F ဖြစ်တဲ့အတွက်ကြောင့် FPU feature ကို စမ်းနိုင်ပါတယ်။ ARM ပရိုဆက်ဆာတွေကနေ လုပ်ဆောင်နိုင်တဲ့ feature တွေကို သူတို့ရဲ့ နာမည်ကြည့်ပြီးသိနိုင်ပါတယ်။ ဥပမာ - ARM7TDMI ဆိုတဲ့ ပရိုဆက်ဆာမှာ 'T' က 'Thumb instruction set architecture' ကိုပြောတာဖြစ်ပါတယ်။ 'D' က 'Debug interface(JTAG)' ပါတာကိုပြောတာပါ။ 'M' က 'fast Multiplier' ပါဝင်လာတာကိုပြောတာဖြစ်ပြီးတော့ 'I' က 'in-circuit emulator' ကိုပါဝင်ခြင်းဖြစ်ပါတယ်။



ပုံ() Cortex-M3 နှင့် Cortex-M4 မိုက်ခရိုပရိုဆက်ဆာ များ၏ တည်ဆောက်မှု ကွာခြားပုံ

ARM Cortex-M3 ပရိုဆက်ဆာတွေမှာ အောက်ပါ feature တွေပါဝင်ကြပါတယ်။

1. Instruction Set Architecture
2. Register Set
3. Processor Operating Modes
4. Interrupts and Processor Reset Sequence
5. Peipelined Architecture and Data Path
6. Memory Address Map



ပုံ() ARM Cortex-M အပေါ်အခြေခံတည်ဆောက်ထားတဲ့ မိုက်ခရိုကွန်ထရို၏ ဖွဲ့စည်းတည်ဆောက်ပုံဖြစ်ပါတယ်။ အပေါ်မှာ ဘောင်ခတ်ပြထားတာက ARM က ဆောက်ထားတဲ့ဖြစ်ပြီး။ အောက်မှာ ဘောင်ခတ်ပြထားတာက Chip manufacturer အပေါ်မူတည်ပြီး ကွာခြားမှုရှိပါတယ်။

Feature label	Description
T	Thumb
D	On-chip debug support
M	Enhanced multiplier
I	Embedded ICE hardware
T2	Thumb2
S	Synthesizable code
E	Enhanced DSP instruction set
J	JAVA support, Jazelle
Z	TrustZone extension
F	Floating point unit
H	Handshake, clockless design for synchronous or asynchronous design

ARM Instruction Set Architecture

ARM ပရိုဆက်ဆာတွေမှာ instruction set နှစ်ခုကို support ပေးထားပါတယ်။ 32 bits ရှိတဲ့ ARM instruction တွေနဲ့ 16 bits ရှိတဲ့ Thumb instruction တွေဆိုပြီးတော့ instruction နှစ်မျိုးကို support ပေးထားပါတယ်။ Instruction တစ်ခုရဲ့ size ဟာ - အဲဒီ instruction ရဲ့ opcode သို့မဟုတ် machine code တွေကိုသိုလှောင်ဖို့ရန်အတွက် လိုအပ်တဲ့ bit အရေအတွက်ဖြစ်ပါတယ်။ Thumb instruction set က ARM instruction တွေရဲ့ တစ်စိတ်တစ်ပိုင်းဖြစ်ပါတယ်။ Thumb Instruction ကိုသုံးရင် ကုန်ကျကျပ်လစ်ပြီးတော့ ARM instruction တွေကိုသုံးရင် ပရိုဆက်ဆာရဲ့ အလုပ်လုပ်နိုင်စွမ်း (execution performance) ကို တိုးတက်စေပါတယ်။

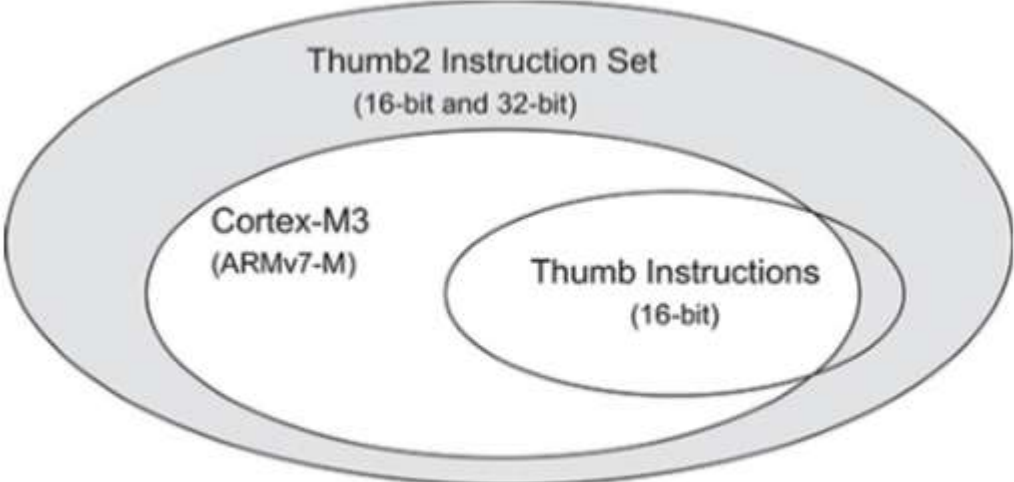
ပရိုဂရမ် ရိုးရိုးလေးတစ်ပုဒ်နဲ့ စဉ်းစားကြည့်ပါမယ်။ Operation ၆ ခုပါဝင်တဲ့ ပရိုဂရမ်တစ်ပုဒ်ဆိုပါတော့ အဲဒီ ပရိုဂရမ်မှာ operation ၃ ခုက Simple Operation ဖြစ်ပြီး ကျန်တဲ့ ၃ ခုက Complex Operation ဖြစ်တယ်ဆိုပါတော့ဗျာ။ Simple Operation သုံးခုကို Assembly instruction သုံးခုနဲ့ရေးထားပြီးတော့ Thumb instruction ကော ARM instruction ကော နှစ်ခုစလုံး support ပေးထားပါတယ်။ နောက် Complex Operation ၃ ခုကိုလည်း Assembly instruction သုံးခုနဲ့ ရေးထားပြီး ARM instruction set က support ပေးထားပါတယ်။ Complex operation တွေထဲက တစ်ခုစီကို assembly instruction နှစ်ကြောင်းသုံးပြီးတော့ thumb instruction set က support ပေးပါတယ်။ ပရိုဂရမ်ထဲမှာ ရေးထားတဲ့ instruction တစ်ကြောင်းကို အလုပ်လုပ်ဖို့ရန်အတွက် one cycle ကြာတယ်လို့ယူဆပါ။ (ဒီအပိုဒ်က ပရိုဂရမ်နဲ့ပတ်သတ်ပြီး ယူဆကြည့်တာဖြစ်ပါတယ်။)

ဒါဆိုရင် စောနက Operation ၆ ခုအတွက် instruction ၆ ခုရှိပါတယ်။ ARM ကော Thumb ကော အတွက် clock speed အတူတူ ပေးထားလိုက်မယ်။ ARM ISA ကိုသုံးတယ်ဆိုရင် အဲဒီ instruction ၆ ခုကို အလုပ်လုပ်ဖို့ရန်အတွက် clock cycle စုစုပေါင်း ၆ ခုကြာမှာဖြစ်ပါတယ်။ Thumb ကိုသုံးတယ်ဆိုရင် တော့

clock cycle စုစုပေါင်း ၉ ခုကြာမှာဖြစ်ပါတယ်။ ဘာလို့ clock cycle ၉ ခုလိုတာလဲဆိုတော့ အပေါ်မှာပြောခဲ့ပါတယ်။ (Complex operation တွေထဲက တစ်ခုစီကို assembly instruction နှစ်ကြောင်းသုံးပြီးတော့ thumb instruction set က support ပေးပါတယ်။) အဲဒါကြောင့်ပါ။

တစ်ဖက်မှာကြည့်မယ်ဆိုရင်လည်း ARM ISA ကိုသုံးမယ်ဆိုရင် အဲဒီ ပရိုဂရမ်ကိုသိမ်းဖို့ရန်အတွက် 24 bytes memory လိုအပ်ပါတယ်။ Thumb ISA ကို သုံးမယ်ဆိုရင်တော့ 18 byte ပဲလိုမှာဖြစ်ပါတယ်။

ဒါကြောင့် ဒီလိုယူဆလို့ရပါတယ်။ ARM ISA ကိုသုံးရင် Execution performance ကောင်းပြီးတော့ Thumb ISA ကိုသုံးရင် Code density ကောင်းပါတယ်။

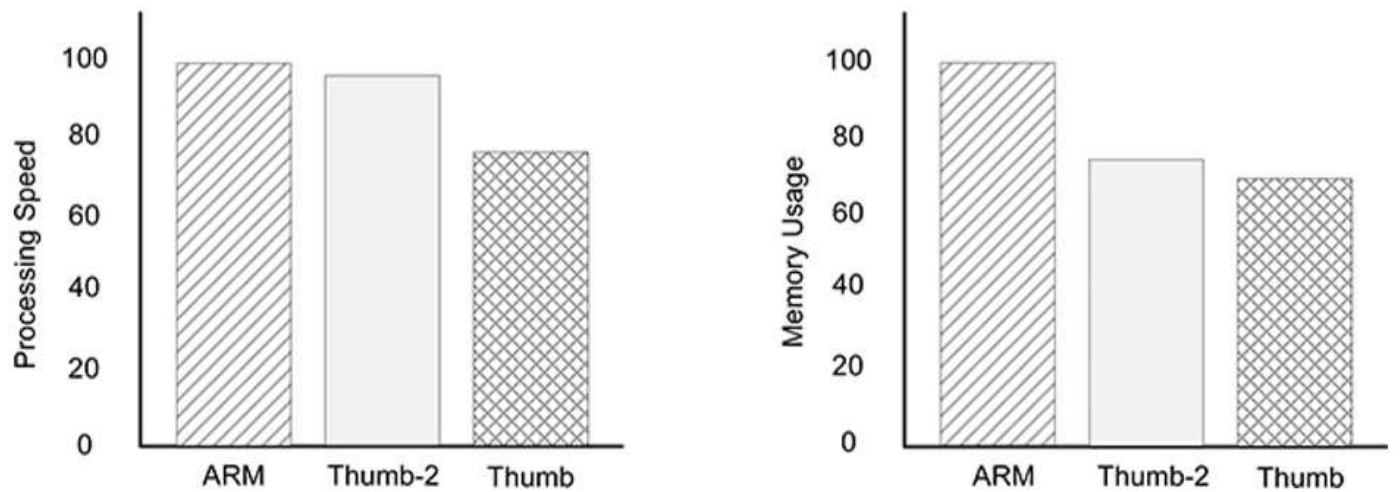


The relationship between ARMv7-M Instruction Set Architecture (ISA) with Thumb2 and the traditional Thumb.

Thumb2 ဟာ Thumb ရဲ့ superset ဖြစ်ပါတယ်။ ပုံမှာပြထားတဲ့အတိုင်းပါဘဲ။ Thumb2 မှာ 32-bit instruction တွေပါဝင်လာပြီး အတွင်းပိုင်းမှာ 16 bit instruction တွေနဲ့ရောထားပါတယ်။ Thumb instruction set နဲ့ လုပ်လို့ရတာတွေအကုန်လုံး Thumb2 မှာအကုန်လုပ်လို့ရပါတယ်။ “Thumb2 ဟာ ARM ISA လိုဘဲ execution performace အနီးစပ်ဆုံး ကောင်းတဲ့အပြင် Thumb ISA လိုမျိုးလဲ code density အတွက် အနီးစပ်ဆုံး ကောင်းပါတယ်။” Thumb2 ISA ဟာ ARM နဲ့နှိုင်းယှဉ်ကြည့်မယ်ဆိုရင် code density မှန်းခြေ ၂၆ ရာခိုင်နှုန်း တိုးတက်တက်မှုရှိလာပြီး 16 bit Thumb architecture နဲ့နှိုင်းယှဉ်ကြည့်တဲ့အခါမှာ performance အရ ၂၅ ရာခိုင်နှုန်းတက်လာတာကို တွေ့ရပါတယ်။

တကယ်တော့ Thumb2 ISA ဆိုတာက Thumb နဲ့ ARM ISA တို့ရောပြီးတည်ဆောက်ထားတာ မဟုတ်ပါဘူး။ ဒါ့အပြင် Thumb2 မှာရှိတဲ့ 32-bit instruction တွေက ARM ရဲ့ 32 bit instruction တွေနဲ့ ကွာခြားမှုရှိပါတယ်။

Instruction Set တွေ့နဲ့ပတ်သတ်တဲ့ အသေးစိတ်အချက်အလက်တွေကို “ARM Architecture Reference Manual” စာအုပ်မှာ ဖတ်လို့ရပါတယ်။ Cortex-M3 ISA အတွက် အသေးစိတ်ကို “ARM v7-M Architecture Reference Manual [ARM(2010)]” ဆိုတဲ့ စာအုပ်မှာဖတ်လို့ရပါတယ်။



ပုံ() Thumb2 ISA ၏ performance အား Thumb ISA နှင့် ARM ISA တို့၏ performance နှင့်

နှိုင်းယှဉ်ထားပုံ