



LAHDEN AMMATTIKORKEAKOULU
Lahti University of Applied Sciences

ARM CORTEX-M-ARKKITEHTUURI

Kehitys ja ominaisuudet

LAHDEN
AMMATTIKORKEAKOULU
Tekniikan ala
Tietotekniikka
Tietokone-elektronikka
Opinnäytetyö
Syksy 2013
Juha-Matti Rintala

Lahden ammattikorkeakoulu
Tietotekniikka

RINTALA, JUHA-MATTI:

ARM Cortex-M-arkkitehtuuri
Kehitys ja ominaisuudet

Tietokone-elektroniikan opinnäytetyö, 36 sivua

Syksy 2013

TIIVISTELMÄ

Työn tarkoituksena oli tutkia Cortex-M-arkkitehtuurin kehitystä ja prosessorien perusominaisuuksia. Työssä käydään läpi ARM LTD:n historia ja se kehitys, joka johti Cortex-M-perheeseen. Cortex-M-perheen suorittimien perusominaisuudet selvitetään toiminnaltaan ja ominaisuuksiltaan yksityiskohtaisesti.

Työssä myös vertaillaan eri Cortex-M-prosessoreja toisiinsa. Suurin osa työstä tehtiin Cortex-M-dokumenttien ja kirjallisuuden pohjalta, jonka sisällöstä kerättiin työhön kaikki tärkeimmät perustiedot. Tärkeimmät tulokset olivat eri Cortex-M-perheen prosessorien ominaisuuksien selvitys ja toiminnan tavat.

Asiasanat: mikroprosessorit, mikrokontrollerit, ARM, Cortex-M

Lahti University of Applied Sciences
Degree Programme in Information Technology

RINTALA, JUHA-MATTI:

ARM Cortex-M Architecture
Development and features

Bachelor's Thesis in Computer Electronics, 36 pages

Autumn 2013

ABSTRACT

The aim of the project was to study the development of Cortex-M architecture and the basic properties of the processors. The work goes through the history of ARM Holdings and the development that led to the Cortex-M series. The basic features and functions of the Cortex-M series processors were examined and all the different Cortex-M processors were compared with each other.

Most of the work was collected from Cortex-M documents and literature. The most important results of this study were the different features and functions of the Cortex-M family of processors.

Key words: microprocessors, microcontrollers, ARM, Cortex-M

SISÄLLYS

1	JOHDANTO	1
2	ARM HOLDINGS PLC:N HISTORIA (ARM) JA ARM ARKKITEHTUURIN KEHITYS	2
2.1	ARM-yhtiön historia	2
2.2	ARM-tuotteet	4
2.3	ARM-arkkitehtuurin kehitys	6
3	CORTEX-M-PERUSTEET	9
3.1	Toimintatilat	9
3.2	Rekisterit	11
3.2.1	R0 – R12: Yleiskäyttöiset rekisterit	12
3.2.2	R13: Pino-osoitin (SP)	12
3.2.3	R14: Linkkirekisteri (LR)	13
3.2.4	R15: Ohjelmalaskuri (PC)	13
3.2.5	PSR: Ohjelmatilarekisteri	14
3.2.6	PRIMASK-, FAULTMASK- ja BASEPRI-rekisterit	15
3.2.7	CONTROL: ohjausrekisteri	16
3.3	Nested Vectored Interrupt Controller (NVIC)	17
3.4	Muistiavaruus	19
3.5	Väylärakenteet	23
3.6	Memory Protection Unit (MPU)	23
3.7	Käskykanta	23
3.8	Keskeytykset ja poikkeukset	24
3.9	Testausrajapinnat	24
4	CORTEX-M PROSESSORIEN VERTAILU	25
4.1	Piivalmistus	25
4.2	Käskykannat	26
4.3	Ytimet	27
4.3.1	Cortex-M0	27
4.3.2	Cortex-M0+	28
4.3.3	Cortex-M1	29
4.3.4	Cortex-M3	30
4.3.5	Cortex-M4	31
5	YHTEENVETO	34

1 JOHDANTO

Työssä tutustutaan ARM-yhtiön prosessoriarkkitehtuurin kehitykseen ja Cortex-M-prosessorien perusteisiin. ARM-arkkitehtuurin alle on yhtiön alkuajoista lähtien noussut suuri perhe erilaisia tuotteita. On sanottu, että ainakin yksi ARM-arkkitehtuurilla oleva prosessori löytyy melkein jokaisesta mobiili-laitteesta.

Työssä tutustutaan tarkemmin Cortex-M-perheen tuotteisiin, joiden perustana oli tuoda uusia ja parempia ominaisuuksia mikrokontrollerimarkkinoille. Vuonna 2004 julkaistu Cortex-M3 oli ensimmäinen prosessori, joka oli varustettuna uudella ARMv7-arkkitehtuurilla. Cortex-M3-perheen suosio ja myynti loivat ARM-yhtiölle uskoa, että se pystyy parantamaan myyntiään mikrokontrollerimarkkinoilla.

Cortex-M3-mallin jälkeen julkaistiin Cortex-M0- ja -M1-perhe, joka perustui vanhempaan ARMv6-arkkitehtuuriin. Nämä tuotteet toivat markkinoille vielä halvempia ratkaisuja mikrokontrollereihin. Viimeisimmät päivitykset perheeseen ovat olleet Cortex-M4- ja Cortex-M0+-mallit. Cortex-M4 oli paranneltu malli Cortex-M3:sta ja Cortex-M0+ optimoitu malli Cortex-M0:sta, johon oli lisätty ominaisuuksia uudemmasta ARMv7-arkkitehtuurista.

Työn tutkintaosuudessa selvitetään Cortex-M-perheen perusteet ja perusosat, jotka löytyvät jokaisesta perheen prosessorista. Viimeisessä luvussa vertaillaan kaikkien perheiden tuotteiden ominaisuuksia ja rakenteita toisiinsa.

Työn taustamateriaalina käytettiin ARM-yhtiön sivuilla suositeltua kirjallisuutta ja dokumentteja. Tärkeimmät kirjalliset lähteet olivat Joseph Yiun kirjoittamat ja Elsevier-julkaisuyhtiön julkaisemat kirjat: The definitive guide to the ARM Cortex-M0 ja The definitive guide to the ARM Cortex-M3. Sähköisistä lähteistä tärkeimpiä olivat ARM-dokumentaatiot eri prosessoreista.

2 ARM HOLDINGS PLC:N HISTORIA (ARM) JA ARM ARKKITEHTUURIN KEHITYS

2.1 ARM-yhtiön historia

ARM-arkkitehtuurin kehitys sai alkunsa Iso-Britanniassa Acorn Computers Ltd:n toimesta 1980-luvun alkupuolella. Ensimmäinen ARM-piiri kehitettiin vuosien 1983 – 1985 välillä. Acorn Computers oli uraauurtava mikrotietokoneiden kehittäjä ja yksi johtavista nimistä Iso-Britannian PC-markkinoilla. Acornin alkuperäinen menestys 80-luvun taitteessa valmistetuilla henkilökohtaisilla tietokoneilla sinetöityi, kun British Broadcasting Corporation, Iso-Britannian julkinen yleisradio ja tuotantoyhtiö, palkkasi yrityksen valmistamaan uuden kotitietokoneen nimeltään BBC Microcomputer. Tämä kone tulisi olemaan osa lähetysyhtiön yleistä tietokoneiden opetuskampanjaa, jota näytettiin televisiossa. (Atack & van Sommaren 1993, luku 1.)

BBC-micro kehitettiin 8-bittisen 6502-prosessorin ympärille. Samaa prosessoria oli käytetty Apple II -tietokoneessa. Koneen julkaisu vuonna 1982 osui juuri kotitietokoneiden nousukauteen Iso-Britanniassa ja ylitti myyntitilastoissa kaikki odotukset. Alkuperäisissä arvioissa yksiköitä myytäisiin parhaimmillaan kymmeniä tuhansia kappaleita, mutta todellisuudessa koneita tultiin myymään melkein 2 miljoonaa kappaletta. Kehittyneimmän MASTER-mallin valmistus lopetettiin vasta 1993. (Atack & van Sommaren 1993, luku 1.)

BBC-mikron menestyksen jälkeen Acorn alkoi tutkia, kuinka siirtyä eteenpäin yksinkertaisesta 6502-mikroprosessorista yritysmaailman markkinoille, jota IBM PC hallitsi. Mutta tuotekehityksessä ongelmaksi tuli, että yksikään markkinoilla olleista 16- tai 32-bittisistä prosessoreista ei ollut tarpeeksi tyydyttävä uuden projektin tarpeisiin. Suurin syy tähän olisi, että minkään ratkaisun siirtonopeudet eivät parantuneet hirveästi 6502:n tarjoamasta nopeudesta. Tämän takia yhtiön kehitystiimi päätyi suunnittelemaan kokonaan uuden suoritinpiirin, joka perustui osalta Berkley RISC -projektin julkaisemiin papereihin ja loisi ympäristön, joka olisi samantyylinen tutun 6502-prosessorin kanssa, mutta toimisi 32-bittisessä ympäristössä. (Atack & van Sommaren 1993, luku 1.)

Tämän suunnittelu- ja kehitysprojektin lopputulos oli ensimmäinen ARM-prosessori nimeltään Acorn RISC Machine, jonka nimeksi myöhemmin muutettiin Advanced RISC Machine. (Atack & van Sommaren 1993, luku 1.)

Ensimmäiset koeversiot ARM1-prosessoreista toimitettiin vuonna 1985, ja ne saavuttivat halutun määritelmän, jossa piisiru sisältäisi enintään 25000 transistoria. ARM1-prosessoria seurasi nopeasti ARM2, joka oli ensimmäinen sarjavalmistusversio tuotteesta, ja se oli julkaistaessa yksi yksinkertaisimmista RICS-prosessoreista maailmassa. Tämä yksinkertaisuus johti tarvittuun pienempään tehonkulutukseen, mutta samalla parempaan suorituskykyyn kuin monilla aikakauden 32-bittisillä suorittimilla, ja kehitys jatkui ARM3-versiolla. Myöhemmin Apple Computers- ja ARM-suorittimet valmistanut VLSI Technology alkoi työskennellä Acornin kanssa uusien ARM-ytimien suunnittelussa. (Atack & van Sommaren 1993, luku 1.)

Vuonna 1990 Acorn irtaannutti kehitystiimensä uudeksi yhtiöksi nimeltään Advanced RISC Machines LTD, josta myöhemmin tuli ARM Holdings. Tämän myötä yhtiö luopui ytimien valmistamisesta ja alkoi lisensoida arkkitehtuuriaan muille valmistajille. Vuonna 1991 ARM julkisti ARM 6 -prosessoriperheen ja VLSI oli ensimmäinen lisensoija. Ajan myötä muutkin valmistajat rupesivat lisensoimaan ARM-prosessorisuunnitelmia; näihin kuuluivat Texas Instruments, NEC, Sharp ja ST Microelectronics. Tämän seurauksena alkoi ARM prosessoreiden laajeneminen mobiilipuhelimiin, kiintolevyihin, käsitietokoneisiin, kotiteatterijärjestelmiin ja moniin muihin kuluttajatuotteisiin. (Atack & van Sommaren 1993, luku 1.)

2.2 ARM-tuotteet

Uuden yritysmallinsa myötä ARM tarjoaa uuden mallin suunnitelmat mikroprosessorivalmistajille ja nämä sitten lisäävät prosessorin pohjan omaan mikrosirusuunnitelmaansa. Prosessorien lisäksi ARM suunnittelee järjestelmän sisäisiä palikoita, lisälaitteita ja piivalmistusprosesseja, joita valmistajat voivat valitessaan käyttää. IP-lisenssien lisäksi ARM tarjoaa ohjelmistojen kehitystyökaluja, kuten C-kääntäjiä, testausliitäntälaitteistoja, laitteistoalustoja ja myös palveluja, kuten konsultointia ja teknisiä kursseja. (Yiu 2011, 20.)

Suosituimpia ARM-tuotteita on ARM7TDMI-prosessori, jota käytetään monissa 32-bittisissä mikrokontrollereissa ympäri maailmaa. Toisin kuin perinteiset 32-bittiset prosessorit, ARM7TDMI tuki kahta eri käskykantaa: toinen oli nimeltään ARM-käskykanta, joka on 32-bittinen, ja Thumb-käskykanta, joka on 16-bittinen. Näitten käskykantojen samanaikainen käyttö lisäsi käännetyn koodin tiheyttä, mikä taas vapautti lisää muistia. Tämän mahdollisti sen, että ARM-prosessoreita pystyttiin käyttämään kaikissa kannettavissa laitteissa, jotka toimivat vähän virtaa kuluttavassa ja rajoitetun muistimäärän ympäristöissä. Tämän takia ARM-prosessorit valitaan ensisijaisesti kannettaviin laitteisiin kuten puhelimiin. (Yiu 2011, 21.)

ARM7TDMI:n menestyksen myötä jatkoi ARM nopeampien ja tehokkaampien prosessorien kehitystä. Tämän kautta syntyi ARM 9 -prosessoriperhe, jota käytetään isossa osassa 32-bittisiä mikrokontrollereita, ja ARM 11 -perhe, jotka olivat hyvin suosittuja älypuhelimissa ja kämmentietokoneissa. Nykyään suurimmassa osaa puhelimissa on ainakin yksi ARM-prosessori sisällään. Uudemmat Cortex-A-prosessorit ovat tehneet tietään moniin kannettaviin internetlaitteisiin, kuten taulutietoneisiin. (Yiu 2011, 22.)

Nopeampien huippumallien lisäksi ARM myös alkoi laajentaa tuoteperhettään alemman luokan tuotteisiin, kuten mikrokontrollereihin. Vuonna 2004 ARM julkisti Cortex-M3-mallin, ja sen valmistus alkoi vuonna 2006. Tämä aloitti uuden suuntauksen mikrokontrollerimarkkinoilla. Kuviossa 1 nähdään, kuinka kehitys muuttui 2004 vuoden jälkeen, kun Cortex-perhe julkaistiin. (Yiu 2011, 22.)

Cortex-M3:n myötä julkaistiin myös halvemmille markkinoille suunniteltu Cortex-M0-malli ja Cortex-M4-malli, joka päivitti Cortex-M3-mallin sisältämään liukulaskennan ja digitaalisen signaalinprosessoinnin. Tänä päivänä on yli 60 lisensoijaa, jotka käyttävät Cortex-M-perheen prosessoreita, ja monet mikrokontrollerivalmistajat toimittavat Cortex-M-pohjaisia ratkaisuja. (Yiu 2011, 22.)

1993	• ARM7
1997	• ARM9TDMI
1998	• ARM920
2000	• ARM922T
2001	• ARM926EJ_S
2002	• ARM1136
2003	• ARM1156 • ARM1176
2004	• Cortex-M3
2005	• Cortex-A8
2006	• Cortex-R4
2007	• Cortex-A9
2008	• Cortex-M1
2009	• Cortex-M0
2010	• Cortex-M4
2011	• Cortex-A15
2012	• Cortex-M0+

KUVIO 1. Suositujen ARM tuotteiden ilmestymisvuosia (Yiu 2011, 21.)

2.3 ARM-arkkitehtuurin kehitys

Vuosien kuluessa ARM jatkoi kehitystyötään prosessorien ja järjestelmän muiden lisäosien kanssa. Suositun ARM7TDMI-prosessorin ja myöhemmin ARM1176TZ(F)-S-prosessorin myötä ominaisuuksien ja lisäyksien kehitystyö johti peräkkäisten ARM-arkkitehtuuriversioiden kehitykseen. Huomattavaa on, että arkkitehtuuriversiot ovat eriäviä verrattuna prosessorien nimiin, esimerkiksi ARM7TDMI perustui ARMv4T-arkkitehtuuriin. (Yiu 2010, 3-4.)

ARMv5E-arkkitehtuuri julkaistiin ARM9E-prosessoriperheen myötä sisältäen ARM926E-S- ja ARM946E-S-prosessorit. Tämä arkkitehtuuri lisäsi "tehostetut" digitaalisen signaalin käsittelykäsyt multimediasovelluksille. (Yiu 2010, 3-4.)

ARM11-prosessoriperheen myötä päivittyi arkkitehtuuri versioon ARMv6. Uusia ominaisuuksia tässä arkkitehtuurissa olivat muistijärjestelmän ominaisuudet ja Single Instruction–Multiple Data (SIMD)-käsyt (laskentayksikkö, joka suorittaa synkronoidusti samaa ohjelmakoodia eri data-alkioille). ARMv6-arkkitehtuuriin kuuluvia prosessoreita olivat ARM1136J(F)-S, ARM1156T2(F)-S ja ARM1176JZ(F)-S. (Yiu 2010, 3-4.)

ARM11-perheen myötä kehittäjät päättivät, että monet näistä uusista teknologioista, kuten optimoitu Thumb-2-käskykanta, olisivat myös yhtä sovellettavissa halpamarkkinoiden mikrokontrolleri- ja autoteollisuuden komponentteihin. Silloin päätettiin myös, että arkkitehtuurin pitäisi olla yhtenäistä hitaimmista kontrollereista korkean tason sovellusprosessoreihin. ARM Holdingsilla oli tarve tuottaa prosessoriarkkitehtuurit, jotka parhaiten soveltuisivat kohdemarkkinoille. Tämä tuottaisi hyvin deterministisen, alhaisen porttimäärän prosessorin hinta-aroille markkinoille ja toiminnoiltaan rikkaan, korkean suorituskyvyn omaavaan prosessorin korkean tason markkinoille. (Yiu 2010, 3-4.)

Taulukossa 1 käydään läpi eri prosessorien nimeämisperusteita, taulukosta myös huomaa, kuinka Cortex-perhe muutti ARM-tuotteita.

TAULUKKO 1. Esimerkkejä ARM-prosessoreista ja niiden arkkitehtuurista (Yiu 2011, 23.)

Proessori	Pääteet prosessorin nimessä	Arkkitehtuuri
ARM7TDMI	T = Thumb käskykannan tuki D = JTAG testaus M = Nopea kertoja I = Sulautettu ICE Moduuli	ARMv4T
ARM920T	T = Thumb käskykannan tuki	ARMv4T
ARM946E, ARM966E	E = Tehostetut digitaalisen signaalin käsittelyn käskyt	ARMv5TE
ARM926EJ-S	E = Tehostetut digitaalisen signaalin käsittelyn käskyt J = Jazelle (JAVA kiihdytin) S = Yhdisteltävissä oleva malli	ARMv5TE
ARM1136J(F)-S	(F) = Valinnainen liukuluku tuki	ARMv6
ARM1176JZ(F)-S	Z = TrustZone turvallisuus tuki	ARMv6
ARM1156T2(F)-S	T2 = Thumb-2 käskykannan tuki	ARMv6
Cortex-A8, Cortex-A9, Cortex-A15	A = Sovellus pohjaiset ratkaisut	ARMv7-A
Cortex-R4(F)	R = Reaaliaikaiset ratkaisut, valinnaisella liukuluku tuella	ARMv7-R
Cortex-M3	M = Mikrokontrolleri	ARMv7-M
Cortex-M1	M = Mikrokontrolleri (FPGA sovellus)	ARMv6-M
Cortex-M0	M = Mikrokontrolleri	ARMv6-M
Cortex-M4	M = Mikrokontrolleri, valinnaisella liukuluku tuella	ARMv7E-M

Seuraavien vuosien aikana ARM laajensi tuoteportfoliotaan tuoden monimuotoisuutta prosessorikehitykseensä, mikä johti arkkitehtuuriversion 7 (v7) syntyyn. Tässä versiossa arkkitehtuurimallit jaettiin kolmeen eri profiiliin (Yiu 2010, 4.):

- A-Profiili: suunniteltu korkean suorituskyvyn sovellusalustoille
- R-Profiili: suunniteltu korkean tason sulautettuihin järjestelmiin, joissa reaaliaikainen suorituskyky on tarpeellista
- M-profiili: suunniteltu syvästi sulautettuihin mikrokontrollerityylisiin järjestelmiin.

Cortex-M-perhe oli ensimmäinen tuote, joka kehitettiin V7-arkkitehtuurille, ja Cortex-M3-perheen prosessorit pohjautuvat ARMv7-M-profiiliin, joka on tarkoitettu mikrokontrollereille. Syvällisemmät kuvaukset profiileista (Yiu 2010, 4.):

- A-Profiili (ARMv7-A): Sovelluspohjaisien prosessorien on pystyttävä ajamaan monimutkaisia sovelluksia, kuten korkean tason sulautettuja käyttöjärjestelmiä, kuten Symbian, Linux ja Windows. Silloin on tarvetta suurille määrille prosessointitehoa, virtuaalisen muistijärjestelmien tukea muistiohjaimilta, valinnaisesti Java-tukea ja turvattua ohjelmien suoritusympäristöä. Hyviä esimerkkejä tästä ovat korkean tason mobiilipuhelimet ja tablettitietokoneet.
- R-Profiili (ARMv7-R): Reaaliaikaisten, korkean suorituskyvyn prosessorien kohdemarkkinat ovat suurimmaksi osaksi sovelluksille, kuten vaikka jarrujärjestelmät ja kiintolevyjen ohjaimet, joissa tarvitaan suurta suoritustehoa, luotettavuutta ja todella nopeaa vasteaikaa.
- M-Profiili (ARMv7-M): Prosessorit on suunnattu halpahintaisille markkinoille, missä suorituskyky on tärkeä ja hinta, virrankulutus, pienet keskeytysajat ja helppokäyttöisyys ovat kriittisiä ominaisuuksia. Tärkeitä myös on teollisuuden ohjaussovellukset ja reaaliaikaiset ohjausjärjestelmät.

3 CORTEX-M-PERUSTEET

Tässä luvussa käsitellään kaikissa Cortex-M-prosessoreissa olevia perusosia ja käsitteitä. Cortex-M-prosessorit perustuvat joko ARMv7- tai ARMv6-arkkitehtuurien pohjalle ja tarkemmin -M-profiiliin. Cortex-M3 oli ensimmäinen perheen prosessoreista, jonka menestyksen myötä päätettiin myös luoda -M-profiili vanhemman ARMv6-arkkitehtuurin pohjalle yksinkertaisempiin mikrokontrollereihin.

3.1 Toimintatilat

ARMv7-M-pohjaisissa Cortex-M-prosessoreissa on kaksi toimintamoodia ja kaksi etuoikeustilaa, kuten nähdään taulukossa 2. Toimintamoodit (säiemoodi ja käsittelijämoodi) päättävät, ajaako prosessori joko normaalia ohjelmaa vai poikkeuksien käsittelijää, kuten keskeytyksiä tai järjestelmän poikkeuksia. Etuoikeustilat (etuoikeutettu tila tai käyttäjätila) tarjoavat mekanismin, joka suojaa muistihakuja pääsemästä tärkeille alueille, ja myös tilat tarjoavat perustason turvamallin. (Yiu 2010, 14.)

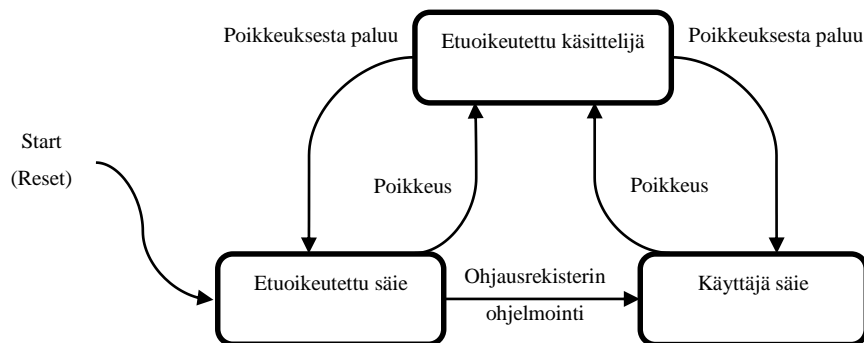
Kun prosessori ajaa pääohjelmaa (säiemoodi), se voi olla joko etuoikeutetussa tilassa tai käyttäjätilassa, mutta poikkeuksien käsittelijät voivat olla vain etuoikeutetussa tilassa. Kun prosessori tulee pois resetistä, se on säietilassa, etuoikeutetulla pääsyoikeuksilla. Etuoikeutetussa tilassa ohjelma pystyy pääsemään käsiksi kaikkiin muistialueisiin (paitsi niihin, joita muistisuojayksikkö on asetettu suojelemaan) ja kaikki käskyt ovat tuettuja. (Yiu 2010, 14.)

Ohjelmat, jotka ovat etuoikeutetussa tilassa, pystyvät vaihtamaan ohjelmat käyttäjätilaan muuttamalla ohjausrekisteriä. Kun poikkeus tapahtuu, prosessori aina palaa etuoikeutettuun toimintatilaan ja palaa takaisin edelliseen tilaansa, kun poikkeuksen käsittelijä lopettaa toimintansa. Käyttäjätilan ohjelma ei pysty muuttamaan tilaa takaisin etuoikeutettuun kirjoittamalla ohjausrekisteriin. Sen pitää mennä poikkeuksen käsittelijän läpi, joka ohjelmoi ohjelmarekisterin vaihtamaan tilansa etuoikeutettuun tilaan. Tämä on havainnollistettu kuviossa 2.

Etuoikeutetun ja käyttäjätilan erittely parantaa todella paljon järjestelmän luotettavuutta estämällä epäluotettavia ohjelmia pääsemästä käsiksi järjestelmän rekistereihin. Jos muistisuojausyksikkö on saatavilla, voidaan sitä käyttää yhdessä etuoikeustilojen kanssa suojelemaan kriittisiä muistipaikkoja, kuten ohjelmia ja käyttöjärjestelmän tietoja. (Yiu 2010, 14.)

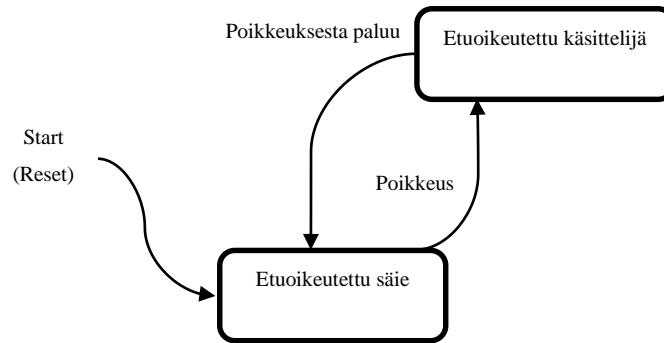
TAULUKKO 2. Käyttäjämoodit ja oikeutetut tilat (Yiu 2010, 15.)

	Etuoikeutetu	Käyttäjä
Ajettaessa poikkeuksen käsittelijää.	Käsittelijämoodi	
Pääohjelma, tai muut ohjelmat	Säiemoodi	Säiemoodi



KUVIO 2. Sallitut tilanvaihdot: ARMv7-M ja ARMv7E-M (Yiu 2010, 15.)

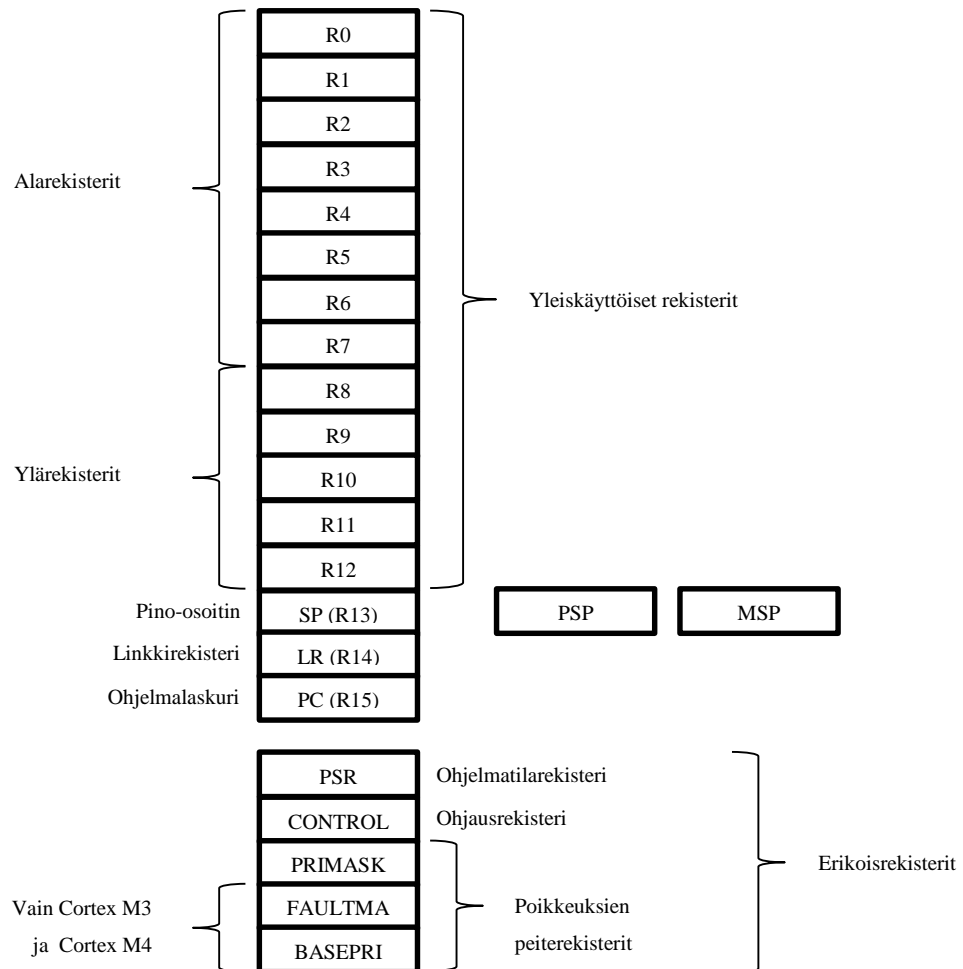
Cortex-M0, -M0 ja -M1-prosessorit perustuvat yksinkertaisempaan rakenteeseen, joka perustuu arkkitehtuuriin ARMv6-M. Näissä prosessoreissa ei ole implementoitu käyttäjätilaa. Kuviossa 3 kuvainnollistetaan, kuinka tämä muutti ohjelmien ajoa.



KUVIO 3. Sallitut tilan vaihdot ARMv6-M (Yiu 2011, 37.)

3.2 Rekisterit

Suorittaakseen tiedon prosessointia ja ohjausta pitää prosessorin sisällä olla tietty määrä tarvittavia rekistereitä. Kun muistissa olevaa tietoa käsitellään, se siirretään muistista rekisteriin rekisteripankissa, joka sen jälkeen suoritetaan prosessorissa ja tarvittaessa kirjoitetaan takaisin muistiin. Kun rekisteripankissa on tarpeellinen määrä rekistereitä, helpottaa se käyttöä ja on C-ystävällinen. C-kääntäjille on helpompaa kääntää ohjelma konekieliseksi koodiksi, jolla on hyvä suorituskyky. Käyttämällä sisäisiä rekistereitä lyhytaikaiseen tiedon talletukseen muistin hakua voidaan vähentää. Suorittimet sisältävät 13 kappaletta yleiskäyttöisiä rekistereitä ja kappalemäärän erikoisrekistereitä, kuten kuvio 4 näyttää. (Yiu 2011, 38.)



KUVIO 4. Rekisterit (ARM Holdings 2010a, 16.)

3.2.1 R0 – R12: Yleiskäyttöiset rekisterit

Rekisterit R0 – R12 on yleiskäyttöisiä. Koska 16-bittisissä Thumb-käskyissä on rajoitettu tila, pystyvät monet Thumb-käskyt vain osoittamaan rekistereihin R0 – R7, joita myös kutsutaan alarekistereiksi. Taas kaikki 32-bittiset Thumb-2-käskyt ja muutamat 16-bittiset Thumb-käskyt voivat käyttää myös rekistereitä R8 – R12, joita kutsutaan ylärekistereiksi. Näiden rekisterien alkuarvot eivät ole määriteltyjä käynnistyksessä. (Yiu 2011, 38.)

3.2.2 R13: Pino-osoitin (SP)

R13 sisältää suorittimen pino-osoittimen. Sitä käytetään pinoon käsiksi pääsyyn PUSH- ja POP-komennoilla. Cortex-M -suorittimet sisältävät kaksi eri pino-

osoitinta jotka ovat pankitettuna rekisteriin SP. Pääasiallinen pino-osoitin (MSP tai SP_main ARM-dokumentaatioissa) on oletuspino-osoitin resetin jälkeen, ja sitä käyttää käyttöjärjestelmien ytimet, poikkeuksien käsittelijät ja kaikki ohjelmat, jotka tarvitsevat etuoikeutettua pääsyä. Prosessipino-osoitinta (PSP tai SP_prosess ARM-dokumentaatioissa) käyttää kantatason ohjelmakoodi (kun poikkeuskäsittelijä ei ole ajossa). (Yiu 2011, 38-39.)

Pino on muistin käsittelymalli. Se on yksinkertaisesti osa järjestelmän muistia, ja pino-osoitinta käytetään sen ohjaukseen ensimmäinen sisään, viimeinen ulostyylisenä puskurina. Yleisin käyttö pinolla on tallentaa rekisterien tiedot ennen kuin tietoa käsitellään ja sen jälkeen palauttaa sisältö takaisin, kun prosessin suoritus on valmista. (Yiu 2011, 38-39.)

Molempien pinojen käyttö ei ole pakollista, yksinkertaiset ohjelmat tulevat toimeen pelkästään MSP:n käytöllä. PSP:tä käytetään yleensä kun suunnitellaan käyttöjärjestelmiä, missä kernelin ja ohjelmien pinojen on oltava erillään. (Yiu 2011, 38-39.)

3.2.3 R14: Linkkirekisteri (LR)

Linkkirekisteriä käytetään paluuosoitteen tallennukseen kun, aliohjelmaa tai funktiota kutsutaan. Näiden suorituksen jälkeen noudetaan LR:ssä oleva paluuosoite ohjelmalaskuriin, jotta kutsuneen ohjelman suoritusta voidaan jatkaa. Poikkeuksien tapauksessa, LR palauttaa myös erikoiskoodin, jota käytetään poikkeuksen paluumekanismissa. (Yiu 2011, 39.)

3.2.4 R15: Ohjelmalaskuri (PC)

R15 sisältää prosessorin ohjelmalaskurin. Se on kirjoitettavissa ja luettavissa. Laskurin luku tuottaa arvon, joka on yleensä neljää suurempi kuin nykyinen suoritusosoite. Tämä tapahtuu, koska prosessori on suunniteltu toimivaksi liukuhihna-periaatteella käskyjen suorituksessa. Kirjoitettaessa laskuriin tapahtuu ohjelman haarautuminen, mutta toisin kuin funktioissa, ei linkkirekisteriä päivitetä. (Yiu 2011, 39.)

3.2.5 PSR: Ohjelmatilarekisteri

PSR on yhdistetty rekisteri, joka tarjoaa tietoa ohjelmien suorituksesta ja ALU:n lipuista. Se koostuu kolmesta Ohjelmantilarekisteristä (PSR):

- Application PSR (APSR)
- Interrupt PSR (IPSR)
- Execution PSR (EPSR)

APSR sisältää matematiikkayksikön (ALU) liput: N (negatiivinen lippu), Z (nolla lippu), C (kertoma- ja jakolippu) ja V (ylivuotolippu). Nämä bitit ovat APSR:n ylimmät bitit, yleisin käyttö lipuille on ehdollisten haarautumien ohjaus. (Yiu 2011, 39-40.)

IPSR sisältää tämänhetkisen suoritettavan keskeytyspalvelurutiinin (ISR) numeron. Jokaisella keskeytyksellä Cortex-M-suorittimissa on uniikki ISR-numero. Tämä auttaa nykyisen keskeytyksen tunnistamista testauksessa, ja se mahdollistaa jaetun poikkeuksien käsittelijän, joka pystyy tiedottamaan sitä käyttäville poikkeuksille, mitä poikkeusta se suorittaa tällä hetkellä. (Yiu 2011, 40.)

ESP sisältää järjestelmän T-bitin, joka osoittaa että järjestelmä on Thumb-suoritustilassa. Koska Cortex-M-prosessorit tukevat vain Thumb-tilaa, on sen oltava asetettuna arvolla 1. (Yiu 2011, 40.)

Yhdessä näihin kaikkiin rekistereihin pääsee yhdistetyn rekisterin xPSR:n kautta. Keskeytyksen tapahtuessa xPSR on yksi niistä rekistereistä, joka tallennetaan pinomuistiin automaattisesti ja palautetaan automaattisesti, kun tapahtumasta palataan. Tämän aikana xPSR käsitellään yhtenä rekisterinä. (Yiu 2011, 40.)

3.2.6 PRIMASK-, FAULTMASK- ja BASEPRI-rekisterit

PRIMASK-, FAULTMASK- ja BASEPRI-rekistereitä käytetään poikkeuksien toiminnan estoon. Taulukko 3 kertoo tarkemmin toiminnan. PRIMASK- ja BASEPRI-rekisterit ovat käytännöllisiä keskeytyksien väliaikaiseen estoon ajastusta vaativissa tärkeissä suorituksissa. (Yiu 2010, 30-31.)

TAULUKKO 3. (Yiu 2010, 31.)

Rekisteri	Kuvaus
PRIMASK	1 bittinen rekisteri, joka kun on asetettu, päästää vain naamioimattoman keskeytyksen ja laitteistohäiriö poikkeuksen toimintaan. Muut keskeytykset ja poikkeukset naamioidaan
FAULTMASK	1 bittinen rekisteri, joka kun on asetettu, päästää vain naamioimattoman keskeytyksen toimintaan. Muut keskeytykset ja poikkeukset naamioidaan
BASEPRI	Rekisteri on 8 bittiin asti(riippuen järjestelmän implementoidusta tärkeystasosta). Se määrittää järjestelmän naamioinnin tärkeystason. Kun asetettuna se poistaa käytöstä kaikki keskeytykset ja poikkeukset jotka ovat asetetun tärkeystason alapuolella. Jos kyseinen rekisteri on asetettuna nolaksi se päästää kaikki lävitse. (Oletusarvo)

3.2.7 CONTROL: ohjausrekisteri

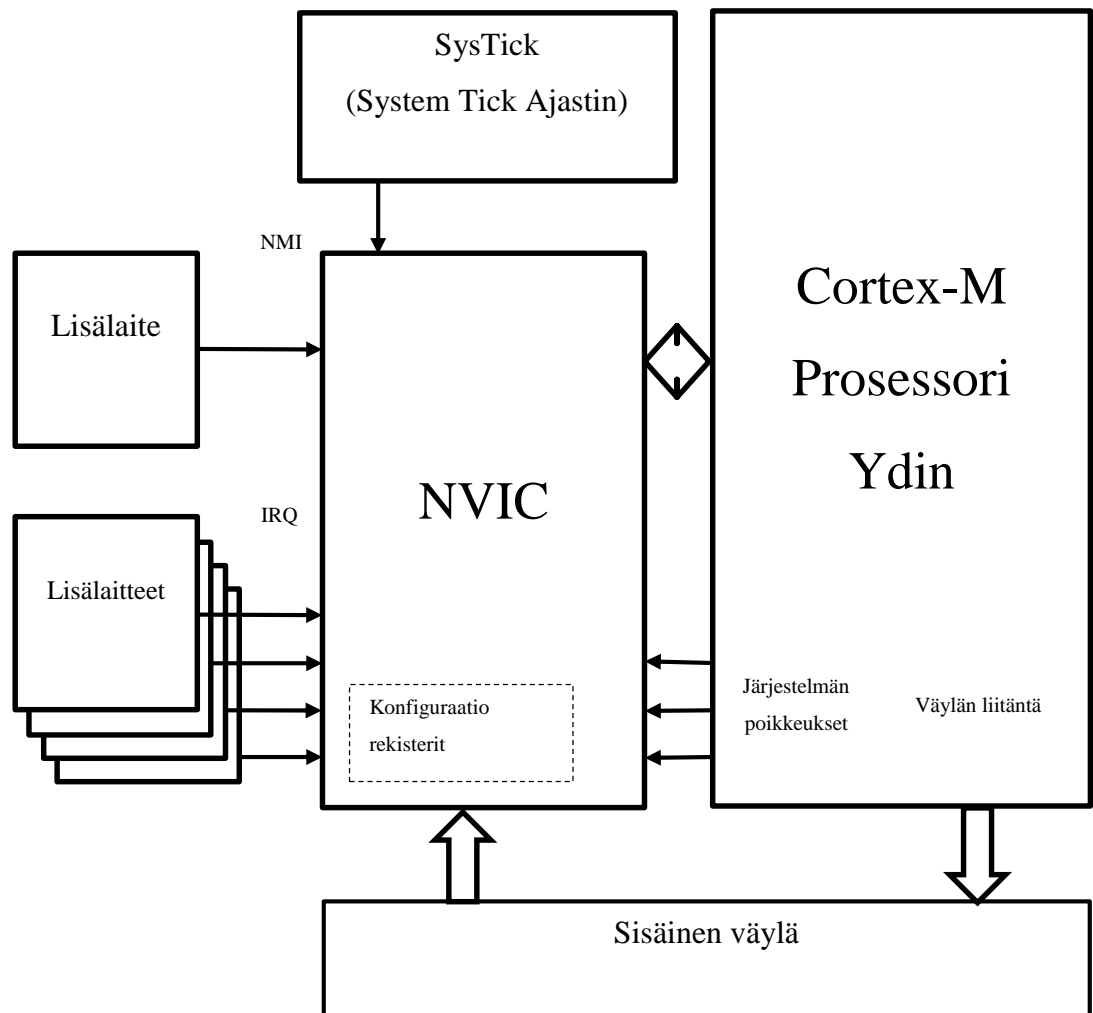
Ohjausrekisteriä käytetään nykyisen suoritustilan määrittämiseen ja käytetyn SP:n valintaan. Se on 2-bittinen rekisteri, jonka toiminnot on selostettu taulukossa 4. Resetin jälkeen käytetään pääasiallista pino-osoitinta (MSP), mutta se pystytään vaihtamaan prosessipino-osoittimeen säiemoodissa. (Yiu 2011, 42.)

TAULUKKO 4. (Yiu 2010, 32.)

Bitti	Toiminto
CONTROL[1]	<p>Pinon tila</p> <p>1 = Vaihtoehtoinen pino käytössä</p> <p>0 = Oletuspino käytössä (MSP)</p> <p>Jos suoritustila on säie tai perustasolla, vaihtoehtoinen pino on PSP. Vaihtoehtoista tilaa ei ole käsittelymoodissa, joten silloin tämän bitin on pakko olla nollattuna, kun prosessori on käsittelymoodissa</p>
CONTROL[0]	<p>0 = Etuoikeutettu säie moodissa</p> <p>1 = Käyttäjätila säie moodissa</p> <p>Jos käsittelymoodissa, prosessori toimii etuoikeutetussa moodissa. Ei käyttäjätila toimintaa ARMv6-M arkkitehtuurin alla: Cortex-M0, -M0+ ja -M1.</p>

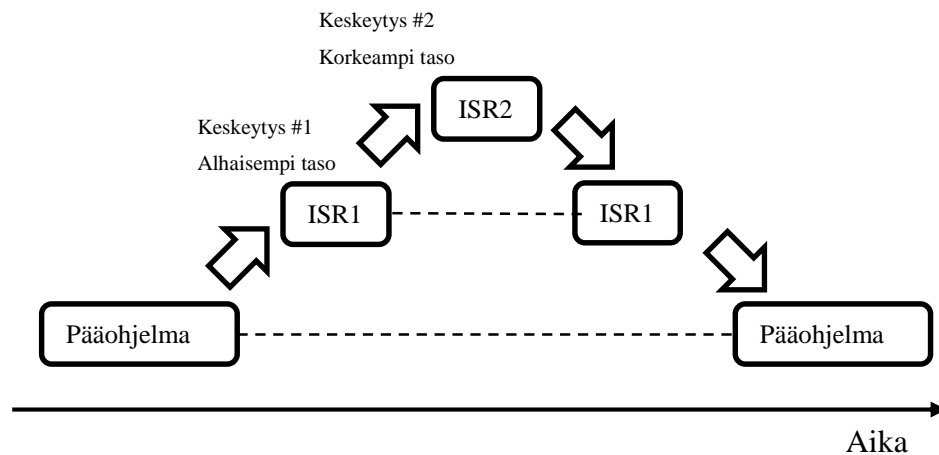
3.3 Nested Vectored Interrupt Controller (NVIC)

NVIC-keskeytysohjain on yksi keskeisistä osista kaikissa Cortex-M-perheen prosessoreissa, ja se tarjoaa niille merkittävät keskeytyksien hallintaominaisuudet. Cortex-M0-, Cortex-M0+- ja Cortex-M1-proessoreissa NVIC tukee 32:ta keskeytystä (IRQ), estämätöntä keskeytystä (NMI) ja useanlaisia järjestelmän poikkeuksia. Cortex-M3- ja Cortex-M4-proessorit laajensivat NVIC-tukea 240 keskeytykseen, yhteen estämätömään keskeytykseen ja isommalle määrälle järjestelmän poikkeuksia. Kuviossa 5 nähdään lohkokaavio, joka näyttää, kuinka eri prosessorin osat yhdistyvät NVIC:iin. (ARM 2013.)



KUVIO 5. NVIC lohkokaavio (ARM 2013.)

Suurin osa NVIC:n asetuksista on ohjelmoitavissa. Sen konfiguraatiorekisterit ovat osana muistikarttaa, ja niihin pääsee helposti käsiksi C-ohjelmoinnissa osoittimilla. CMSIS-kirjasto myös tarjoaa erilaisia avustusfunktioita tehden keskeytyksien hallinnasta helpompaa. NVIC:n sisällä jokaiselle keskeytyslähteelle annetaan tärkeysjärjestys, joillain järjestelmän poikkeuksilla ja estämättömällä keskeytyksellä on lukittu tärkeys taso ja muilla on ohjelmoitava taso. Antamalla eri tärkeysjärjestyksiä jokaiselle keskeytykselle osaa NVIC tukea suoraa keskeytysten suoritusta ilman mitään ohjelmiston väliintuloa. (ARM 2013.)



KUVIO 6. Keskeytyksen toiminta (ARM 2013.)

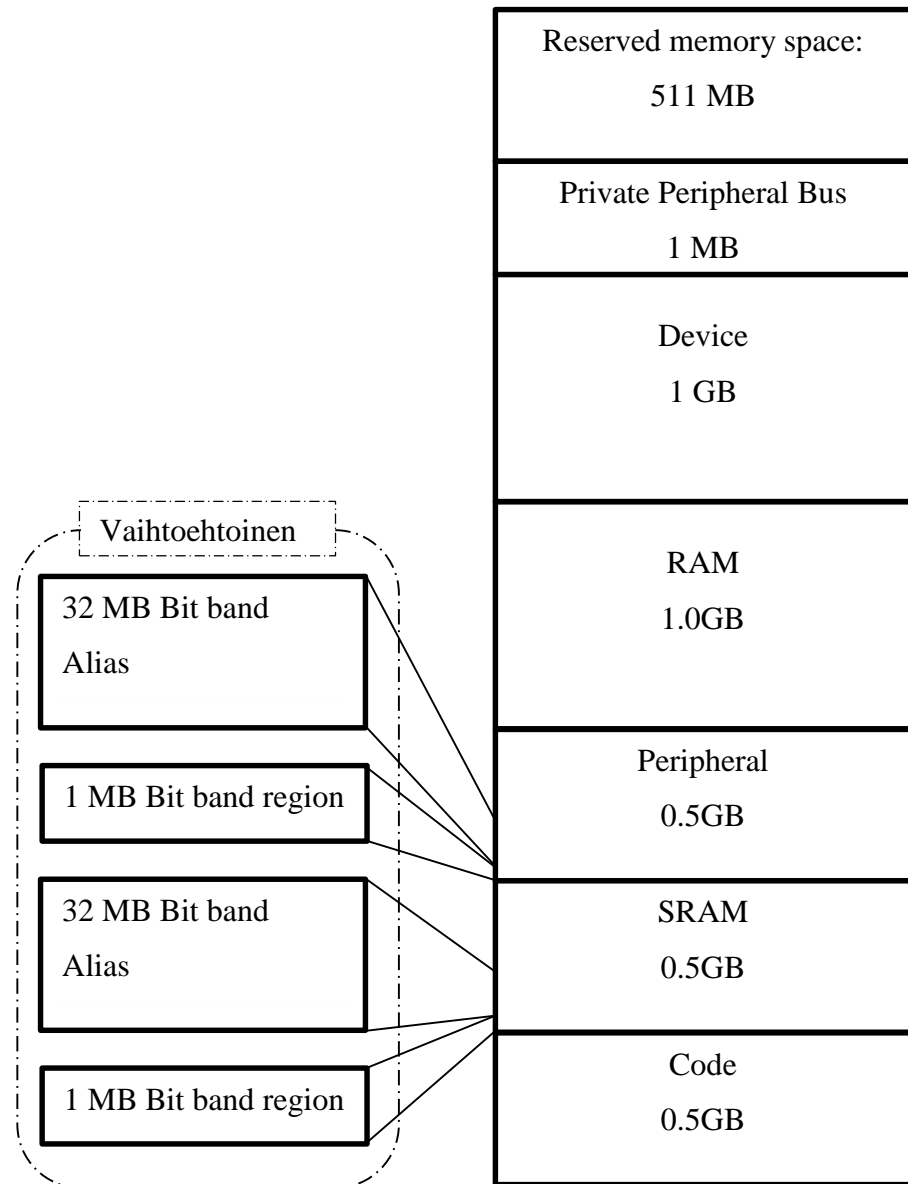
Arkkitehtuuri tarjoaa 8-bittistä tärkeysjärjestystasoa jokaiselle ohjelmoitavalle keskeytykselle tai poikkeukselle. Vähentäkseen porttien määrää prosessorissa vain osa näistä rekistereistä käytetään hyödyksi. Cortex-M0-, -M0+- ja -M1-prosessoreissa (ARMv6-M-arkkitehtuuri) hyödynnetään 4 ohjelmoitavaa tasoa. Cortex-M3- ja Cortex-M4-prosessoreissa (ARMv7-M-arkkitehtuuri) on mahdollista käyttää 8 ja 256 tason väliltä keskeytyksiä. Keskeytystasojen toiminta on esillä kuviossa 6. Helpottaakseen Cortex-M-prosessorien käyttöä käytössä on pinopohjainen poikkeusmalli. Poikkeuksen tapahtuessa viedään osa rekisterien sisällöstä talteen pinoon. Nämä rekisterit palautetaan alkuperäisiin arvoihinsa, kun

poikkeuksen käsittely loppuu. Tämä mahdollistaa poikkeuksien helpomman kirjoituksen normaalina C-funktiona ja vähentää ohjelman viemää suoritusaikaa piilotettujen keskeytyskäskyjen suorituksessa. (ARM 2013.)

Lisäksi Cortex-M-prosessorit käyttävät vektoritaulua, joka sisältää jokaisen funktion osoitteen, jota ollaan suorittamassa missä tahansa valitussa keskeytyskäsittelijässä. Hyväksyttäessä keskeytyksen hakee prosessori osoitteen vektoritaulusta. Tämä vähentää ohjelman käyttämää suoritusaikaa vähentäen myös latenssia keskeytyksen ajossa. Myös erilaisia optimointitekniikoita käytetään Cortex-M-suorittimissa, jotka vielä parantavat keskeytyksien suoritusta ja järjestelmän toimintaa. (ARM 2013.)

3.4 Muistiavaruus

Cortex-M-prosessoreissa on 4 gigatavua muistiosoitetilaa. Tälle muistitilalle on arkkitehtuurillisesti esimääritelty muistikartta, joka on esillä kuviossa 7. Tämä antaa helpon pääsyn sisäänrakennetuille oheislaitteille, kuten keskeytysohjaimelle ja testausohjaimelle suoraan yksinkertaisten muistikomentojen kautta. Tämän takia monet järjestelmän ominaisuudet ovat käytettävissä suoraan C-ohjelmakoodilla. 4 gigatavun muistiavaruus jaetaan erikokoisiin alueisiin, kuten kuviossa 2 näytetään. Cortex-M-prosessorien sisäinen väylärakenne on optimoitu tämän muistimallin käyttöön. Tämän lisäksi malli mahdollistaa eri muistialueiden käytön eri tavoilla. Datamuisti voidaan määrittää koodialueille ja ohjelmakoodia voidaan suorittaa ulkoiselta RAM-alueelta. Järjestelmätason muistialue sisältää keskeytysohjaimen ja testauskomponentit, joilla on lukitut osoitteet. Tämän takia ohjelmien siirto eri Cortex-tuotteiden välillä on paljon helpompaa. (Yiu 2010, 81)



KUVIO 7 Muistimalli (ARM 2010a, 25.)

Koodialue

Koodi alueen koko on 512 megatavua. Sitä käytetään pääosin koodin tallentamiseen, se sisältää poikkeusvektoritaulukon joka on osana ohjelma levykuvaa. Voidaan myös käyttää datan tallentamiseen. (Yhteys RAM:iin). (Yiu 2011, 103-106.)

SRAM-alue

SRAM-alue sijaitsee seuraavassa 512 megatavussa muistiavaruutta. SRAM-alue on pääosin datan tallentamiseen, sisältää myös pinon. SRAM-alue pystyy myös tallentamaan ohjelmakoodia. Jossain tapauksissa voidaan kopioida ohjelmakoodia hitaalta ulkoiselta muistilta nopeampaan SRAM:iin ja suorittaa sitä sieltä. Vaikka tätä aluetta nimitetään SRAM-alueeksi, voi siinä käytettävä muisti olla mitä tahansa, kuten SRAM, SDRAM ja uudemmat teknologiat. (Yiu 2011, 103-106.)

Laitealue

Laite-alue koostuu myös kahdesta 512 megatavun lohkoista, jotka ovat lisälaitteiden ja liitäntöjen käytössä. Näitäkään alueilta ei voi suorittaa ohjelmia mutta käy alueet käyvät yleisen tiedon tallentamiseen. Ainoa ero näiden kahden muistilohkojen välillä on niiden attribuutit. (Yiu 2011, 103-106.)

Bit Band -operaatiot

Bit Band -alueet tukevat yhden luku- tai kirjoitus-operaation yhteen databittiin. Nämä alueet ovat tuettuina kaikissa muissa paitsi Cortex-M1 prosessoreissa. Näihin kahteen alueeseen pääsee käsiksi samalla tavalla kuin normaaliin muistiin tai niihin voi myös päästä käsiksi erillisten bit-band alias-alueiden kautta. Näillä operaatioilla saadaan nopeammin tehtyä yksittäisten bittien vaihto muistissa. (Yiu 2011, 103-106.)

RAM-alue

RAM-alue koostuu kahdesta 512 megatavun lohkoista joista tulee yhteensä 1 gigatavu tilaa. RAM-alue on pääosin datan tallennukseen ja useimmissa tapauksissa sitä voidaan käyttää 1 gigatavun yhtenäisenä muistialueena. Sitä

voidaan käyttää ohjelmakoodin suorittamiseen. Ainoa ero näiden kahden muistilohkojen välillä on niiden attribuutit, jotka voivat aiheuttaa erilaisuuksia, jos aluetta käytetään järjestelmätason välimuistina. (Yiu 2011, 103-106.)

Private Peripheral Bus (PPB)-alue

PPB-muistiavaruus on varattu prosessorin sisäisille lisälaitteille, kuten NVIC:lle. Se on kooltaan 1 megatavu, ja ohjelmien suoritus tältä alueelta ei ole sallittua. Tämän muistialueen sisällä sijaitsee myös System Control Space, joka sisältää keskeytyksen ohjaus-, järjestelmän ohjaus- ja testauksen ohjausrekisterit. Myös NVIC:n rekisterit ovat osana SCS:n muistiavaruutta. Se sisältää myös vaihtoehdoisen ajastimen SysTick. (Yiu 2011, 103-106.)

Varattu alue

Viimeinen osa muistiavaruutta on 511 megatavun varattu muistiavaruus. Tämä alue voi olla varattu joissain mikrokontrollereissa valmistajakohtaisiin käyttötarkoituksiin. (Yiu 2011, 103-106.)

Vaikka Cortex-M-perheen prosessoreissa on tällainen muistikartta, on muistin käyttö hyvin joustavaa. Siinä voi olla useita SRAM-muistilohkoja SRAM- ja CODE-alueille. Ohjelmakoodia voidaan myös suorittaa ulkoisilta muistikomponenteilta RAM-alueella. Valmistajat voivat myös lisätä omia järjestelmätason muistiominaisuuksia. (Yiu 2011, 103-106.)

Tyypilliset lisät:

- Flash-muisti (ohjelmakoodille)
- sisäinen SRAM (datalle)
- sisäiset lisälaitteet
- ulkoinen muistiliitäntä (ulkoisille muisteille ja lisälaitteille)
- liitännät muille ulkoisille lisälaitteille.

Lisälaite alue

Lisälaite alue on kooltaan 512 megatavua. Alue on lisälaitteiden käytössä, mutta sitä voidaan myös käyttää datan tallentamiseen. Ohjelman suorittaminen tältä alueelta ei ole sallittua. (Yiu 2011, 103-106.)

3.5 Väylärakenteet

Väylärakenteissa suurimmat erot näkyvät Cortex-M-perheen prosessoreissa. ARMv7-perhe oli suunniteltu Harvard-prosessoriarkkitehtuurin mukaan, mikä tarkoittaa sitä, että prosessorin sisällä on fyysisesti eri väylät muistille ja signaaleille, kuten käskyt ja data. ARMv6-pohjalle tehdyt prosessorit ovat Von Neumann -arkkitehtuurin pohjalla, jossa on yhteinen väylä muistille ja signaaleille. Harvard-arkkitehtuurin prosessorit ovat suorituskyvyltään nopeampia, mutta ARMv6-pohjaiset suorittimet ovat yksinkertaisempia ja halvempia valmistaa. (Yiu 2010, 17.)

3.6 Memory Protection Unit (MPU)

Cortex-M3, M4 ja M0+ sisältävät mahdollisuuden vaihtoehdoiselle muistinsuojausyksikölle. Tämän avulla pystytään luomaan sääntöjä eri toimintatilan luvuille. Kun sääntöä rikotaan, luodaan poikkeus ja poikkeuksen käsittelijä pystyy analysoimaan ongelman ja korjaamaan sen, jos mahdollista. MPU:n käyttö helpottaa käyttöjärjestelmien luontia ja mahdollista muistien suojausta vahinko ylikirjoituksilta. Yleensä se tekee sulautetuista järjestelmistä vakaampia ja luotettavampia. Tämä on vaihtoehtoinen ominaisuus, joka liitetään prosessoriin valmistus suunnitelmaan ennen piirinvalmistusta. (Yiu 2010, 18.)

3.7 Käskykanta

Cortex-M-perheen prosessorit tukevat Thumb-2-käskykanta. Tämä on tärkein kehitysominaisuus Cortex-M-perheessä, koska se mahdollistaa 32-bittisten ja 16-bittisten käskyjen yhtäaikaisen suorituksen, mikä parantaa koodin tiheyttä ja suorituskykyä. Edellisissä ARM-prosessoreissa käytettiin kahta eri käskykanta:

32-bittistä ARM-tilaa ja 16-bittistä Thumb-tilaa. Näiden tilojen välillä pitää aina tehdä vaihto, jos halutaan käyttää toista käskykantaa. Thumb-2-käskykannan myötä kaikki operaatiot voidaan suorittaa Thumb-tilassa, Cortex-M-perheen prosessorit eivät edes tue ARM-tilan käyttöä. Jopa keskeytykset ajetaan Thumb-tilassa (ennen ARM-ytimen piti vaihtaa ARM-tilaan keskeytyksen suorittamisen ajaksi). Huomioitava on, että eri Cortex-M-ytimissä on tuettuna vain tietty osa Thumb-2-käskyistä, koska tämä yksinkertaistaa suorittimen rakennetta. (Yiu 2010, 18.)

3.8 Keskeytykset ja poikkeukset

Cortex-M-perheen prosessorit toivat uuden poikkeus mallin ARMv7-M-arkkitehtuurin myötä. Tämä malli mahdollisti tehokkaan poikkeuksien hallinnan. Uudet ominaisuudet implementoitiin NVIC:ssä. Ulkoisten keskeytyksien lisäksi tuetaan myös sisäisiä poikkeuksien lähteitä, kuten järjestelmän vian hallintaa. (Yiu 2010, 19-20.)

3.9 Testausrajapinnat

Cortex-M-prosessoreihin voidaan implementoida täysinäinen rautatason testausratkaisu, joka tarjoaa pääsyn järjestelmän toiminnan tarkkailuun joko JTAG-portin kautta tai 2-pinnin Serial Wire Debug (SWD) -portin kautta. ARMv7-perheen prosessoreille on myös tarjolla Embedded Trace Macrocell –ominaisuus, joka mahdollistaa käskyjen seuraamisen. Kaikkien normaalien testausomaisuuksien lisäksi nämä sisältävät Flash Patch- and Breakpoint-yksikön jolla pystytään luomaan yksinkertaisia keskeytysfunktioita tai siirtämään käskyjen haku Flash-muistilta SRAM-muistiin. (Yiu 2010, 21-22.)

4 CORTEX-M PROSESSORIEN VERTAILU

4.1 Piivalmistus

Piirien valmistaja saavat ARM-prosessorien mallin yhtiöltä. Tässä muodossa heillä on mahdollisuus tehdä arkkitehtuuritason optimointeja ja laajennuksia. Tämä antaa valmistajille parhaan mahdollisen pohjan, jolla voidaan suunnitella uusi suoritin sellaisilla ominaisuuksilla kuin he haluavat. Taulukossa 5 havainnollistetaan eri suorittimien valmistusmahdollisuudet.

Tärkeimmät Cortex-M perheen lisäkomponentit ovat seuraavat (Wikipedia 2013):

- SystemTick ajastin: 24-bittinen ajastin, joka laajentaa sekä prosessorin että NVIC:n toimintaa; vaikka tämä on vaihtoehtoinen komponentti, se harvoin jätetään pois piiristä
- Bit Band -operaatiot, yksittäisten bittien muunnokset muistiosoitteissa
- Memory Protection Unit (MPU), muistinsuojayksikkö.

TAULUKKO 5. ARM Cortex-M lisäkomponentit (Wikipedia 2013)

Suoritin	SysTick Ajastin	Bit Band	MPU
Cortex-M0	Vaihtoehtoinen	Vaihtoehtoinen	Ei
Cortex-M0+	Vaihtoehtoinen	Vaihtoehtoinen	Vaihtoehtoinen
Cortex-M1	Vaihtoehtoinen	Ei	Ei
Cortex-M3	On	Vaihtoehtoinen	Vaihtoehtoinen
Cortex-M4	On	Vaihtoehtoinen	Vaihtoehtoinen

4.2 Käskykannat

Cortex-M0, -M0+ ja -M1 käyttävät ARMv6-arkkitehtuuria, Cortex-M3 ARMv7-arkkitehtuuria ja Cortex-M4 ARMv7E-M-arkkitehtuuria. Kaikki arkkitehtuurit ovat ylöspäin yhteensopivia, joten uusien versioiden myötä mitään vanhaa ei ole muutettu mutta uusia käskyjä on tuotu käskykantaan. Tämä osoitetaan taulukossa 6. Suurin muuttuva osa arkkitehtuureissa on laitteistokertolaskun toteutus.

TAULUKKO 6. ARM Cortex-M käskykannat (Wikipedia 2013)

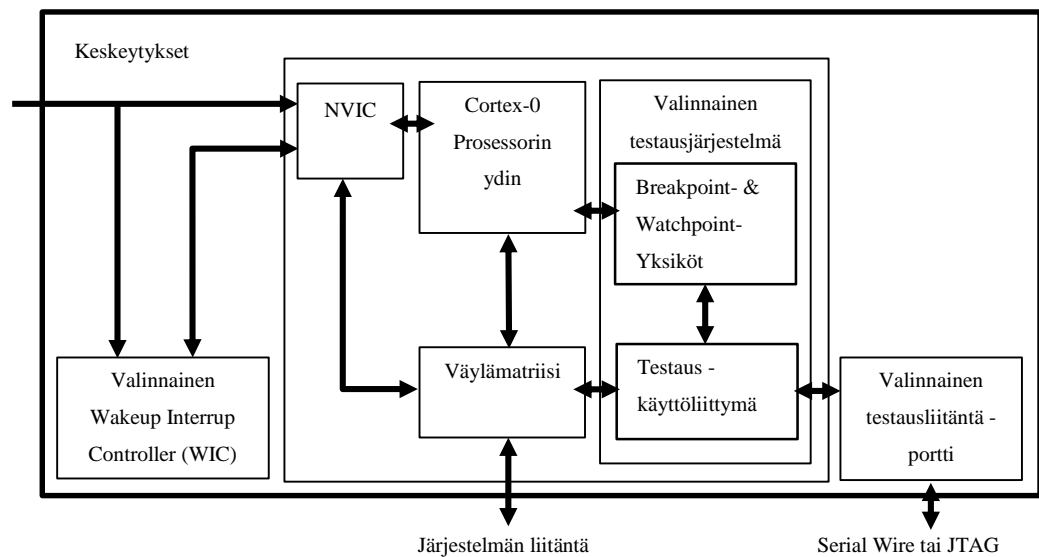
Proessori	Thumb	Thumb-2	Laitteisto kertolasku	Laitteisto jakolasku	Saturaatio laskenta	DSP laajennus	Liukuluku	ARM Arkkitehtuuri	Ydin Arkkitehtuuri
Cortex-M0	Suurin osa	Osa	1 tai 32 kellojakson	Ei	Ei	Ei	Ei	ARMv6-M	Von Neumann
Cortex-M0+	Suurin osa	Osa	1 tai 32 kellojakson	Ei	Ei	Ei	Ei	ARMv6-M	Von Neumann
Cortex-M1	Suurin osa	Osa	3 tai 33 kellojakson	Ei	Ei	Ei	Ei	ARMv6-M	Von Neumann
Cortex-M3	Kaikki	Kaikki	1 kellojakson	Kyllä	Kyllä	Ei	Ei	ARMv7-M	Harvard
Cortex-M4	Kaikki	Kaikki	1 kellojakson	Kyllä	Kyllä	Kyllä	Vaihtoehto	ARMv7E-M	Harvard

4.3 Ytimet

Tässä luvussa vertaillaan, millaisia eri Cortex-M-perheen suorittimet ovat ominaisuuksiltaan toisiinsa verrattuna. Kuvioissa 8, 9, 10 ja 11 ovat eri suorittimien ytimet kuvattuna lohkokaavioina ja näistä voidaan huomata kuinka paljon yksinkertaisempia rakenteeltaan ARMv6-teknologian prosessorit ovat. Taulukot 7, 8, 9 ja 10 tuovat esille eri suorittimien pääominaisuudet.

4.3.1 Cortex-M0

Cortex-M0-ydin on optimoitu käyttämään mahdollisimman pientä piikiekon kokoa joten siitä valmistetut piirit ovat halvimpia.



KUVIO 8. Cortex-M0 ydin (ARM 2009, 14.)

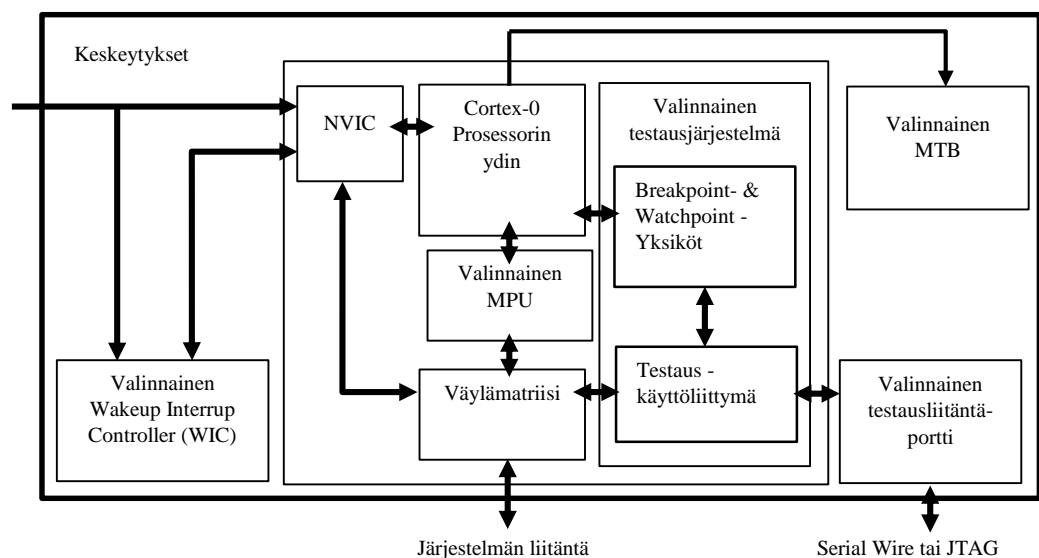
TAULUKKO 7. Cortex-M0 Ominaisuudet (ARM 2013.)

ISA tuki	Thumb [®] / Thumb-2 alijoukko
Pipeline	3-Osainen

Suorituskyky	2.33 CoreMarks/MHz*
Suorituskyky	0.84 / 0.99 / 1.21 DMIPS/MHz**
Keskeytykset	Estämätön keskeytys (NMI) + 1 - 32 fyysistä keskeytystä
Bittien muokkaus	Bit banding alue voidaan tehdä Cortex-M System Design Kitin avulla
Tehostetut käskyt	Laitteistotason 1-kellojakson kertolasku vaihtoehtoisena
Testaus	Vaihtoehtoinen JTAG tai Serial-Wire Debug Portit

4.3.2 Cortex-M0+

Cortex-M0+ on optimoitu versio Cortex-M0:sta. Sen putki vähennettiin kolmesta suorituksesta kahteen, mikä vähentää energian käyttöä. Lisäyksenä alkuperäiseen testausarkkitehtuuriin saa ytimeen mahdollisen MTB-ominaisuuden. Suoritin sai myös piivalmistusmahdollisuuksiksi MPU:n ja vektoritaulukkosiirot.



KUVIO 9. Cortex-M0+ ydin (ARM 2012, 10.)

TAULUKKO 8. Cortex-M0+ Ominaisuudet (ARM 2013.)

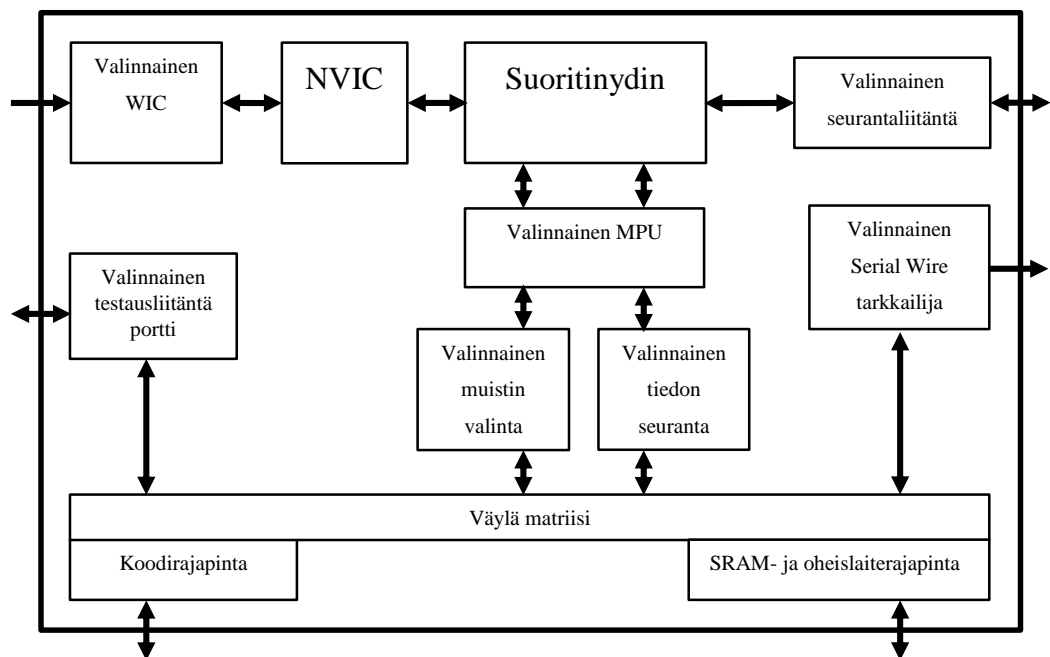
ISA tuki	Thumb [®] / Thumb-2 alijoukko
Pipeline	2-Osainen
Suorituskyky	2.42 CoreMarks/MHz
Suorituskyky	0.93 / 1.08 / 1.31 DMIPS/MHz
Muistin suojaus	Vaihtoehtoinen 8-alueen MPU
Keskeytykset	Estämätön keskeytys (NMI) + 1 - 32 fyysistä keskeytystä
Bittien muokkaus	Bit banding alue voidaan tehdä Cortex-M System Design Kitin avulla
Tehostetut käskyt	Laitteistotason 1-kellojakson (32x32) kertolasku vaihtoehtoisena
Testaus	Vaihtoehtoinen JTAG tai Serial-Wire Debug Portit
Seuranta	Vaihtoehtoinen Micro Trace Buffer

4.3.3 Cortex-M1

Cortex-M1 suunniteltiin pieneksi prosessoriksi, joka pystyttäisiin liittämään ohjelmoitavaa logiikkaa sisältävään FPGA-piiriin. Pinta-alaltaan se on pienin Cortex-M-ratkaisu. Valmistajista Actel, Altera ja Xilinx tukevat Cortex-M1-piirejä tuotteissaan.

4.3.4 Cortex-M3

Cortex-M3 oli ensimmäinen tuote, joka valmistettiin ARMv7-M-arkkitehtuurin perustalta, jonka suosion pohjalta kehitettiin muutkin -M-arkkitehtuurin suorittimet.



KUVIO 10. Cortex-M3 ydin (ARM 2010a, 11.)

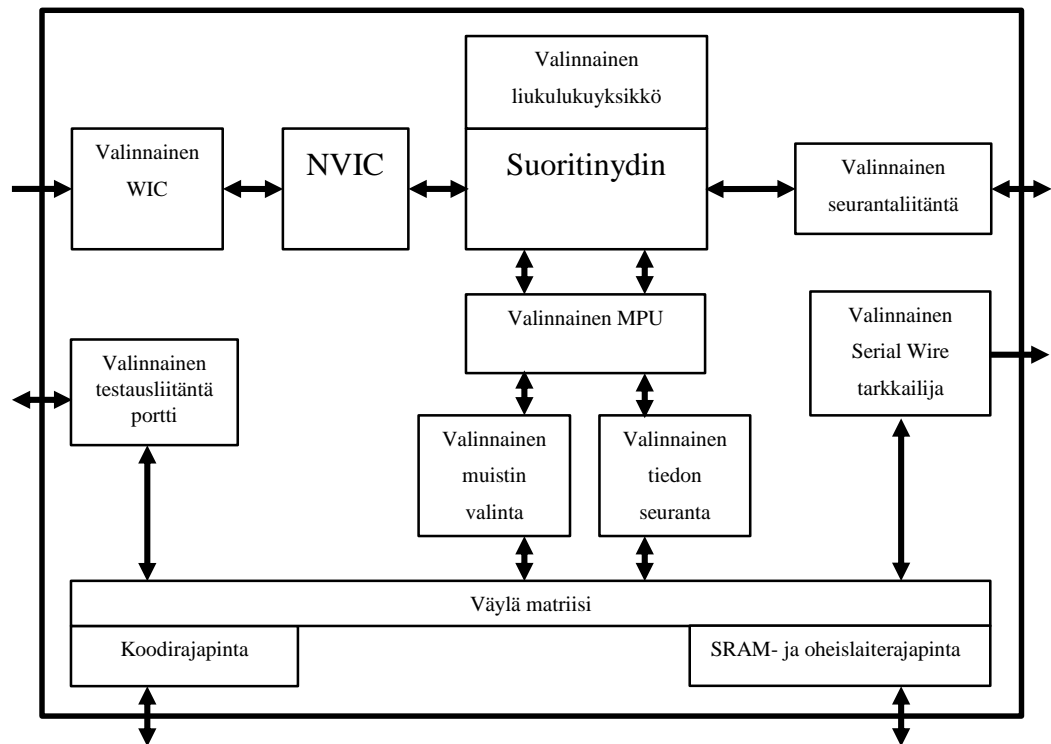
TAULUKKO 9. Cortex-M3 Ominaisuudet (ARM 2013.)

ISA tuki	Thumb [®] / Thumb-2
Pipeline	3-Osainen
Suorituskyky	3.32 CoreMark/MHz
Suorituskyky	1.25 / 1.50 / 1.89 DMIPS/MHz

Muistin suojaus	Vaihtoehtoinen 8-alueen MPU
Keskeytykset	Estämätön keskeytys (NMI) + 1 - 240 fyysistä keskeytystä
Keskeytystasot	8 – 256 Prioriteettitasoa
Bittien muokkaus	Sisällytetyt käskyt ja Bit Banding alueet
Tehostetut käskyt	Laitteistotason jakolasku (2-12 kellojaksoa), yhden kellojakson(32x32) kertolasku, saturoidun laskennan tuki.
Testaus	Vaihtoehtoinen JTAG tai Serial-Wire Debug Portit
Seuranta	Vaihtoehtoinen käskyjen, tiedon ja laitteiston seuranta

4.3.5 Cortex-M4

Konseptiltaan Cortex-M4 on -M3, johon on lisätty digitaalisen signaalin käsittelyn käskyt ja vaihtoehtoinen liukulukuyksikkö. Jos prosessori sisältää liukulukuyksikön, tunnetaan se nimellä Cortex-M4F, muuten se on Cortex-M4.



KUVIO 11. Cortex-M4 ydin(ARM 2010b, 11.)

TAULUKKO 10. Cortex-M4 Ominaisuudet (ARM 2013.)

ISA tuki	Thumb [®] / Thumb-2
DSP laajennus	Kyllä
Liukuluku yksikkö	IEEE 754 standardia tukeva yksikkö
Pipeline	3-Osainen + haarautumien ennustus
Suorituskyky	3.40 CoreMark/MHz
Suorituskyky	Ilman FPU yksikköä 1.25 / 1.52 / 1.91 DMIPS/MHz FPU yksikön kanssa 1.27 / 1.55 / 1.95 DMIPS/MHz
Muistin suojaus	Vaihtoehtoinen 8-alueen MPU

Keskeytykset	Estämätön keskeytys (NMI) + 1 - 240 fyysistä keskeytystä
Keskeytystasot	8 – 256 Prioriteettitasoa
Bittien muokkaus	Sisällytetyt käskyt ja Bit Banding alueet
Tehostetut käskyt	Laitteistotason jakolasku (2-12 kellojaksoa), yhden kellojakson (32x32) kertolasku, saturoidun laskennan tuki.
Testaus	Vaihtoehtoinen JTAG tai Serial-Wire Debug Portit
Seuranta	Vaihtoehtoinen käskyjen, tiedon ja laitteiston seuranta

5 YHTEENVETO

Työn tavoitteena oli tutkia ARM Cortex-M-perheen suorittimia ja niihin johtanutta kehitystä. Suurin osa selvityksessä meni eri perusominaisuuksien selvitykseen prosessoriperheissä. Cortex-M3- ja Cortex-M4-prosessorit perustuivat ARMv7-M-arkkitehtuurin ja sisälsivät itsessään enemmän ominaisuuksia kuin ARMv6-M-arkkitehtuuriin perustuvat Cortex-M0, Cortex-M1 ja Cortex-M0+.

ARM-arkkitehtuurin kehitys alkoi jo 80-luvulta, joten suorittimien kehityksestä on paljon materiaalia, jota piti käydä lävitse. Työn tarkoitus oli luoda (enemminkin) tutkielma Cortex-M-perheen perusominaisuuksista, joten luvun 2 tiedoissa oli paljon mitä pystyi tiivistämään, niin että se sisältäisi vain tärkeimmät kohdat pohjana seuraaville luvuille.

Perusominaisuuksia selvitettäessä oli taas ongelmana se, mitä valita työhön kirjoitettavaksi. Halusin tuoda esiin kaikkien prosessorien yhteiset perusominaisuudet esiin samalla olematta keskittymättä yhteen tiettyyn malliin ja sen kaikkiin erikoisominaisuuksiin. Mielestäni sain hyvin perusteista tiivistettyä kaikki peruskomponentit, niiden toiminnat suorittimissa ja ominaisuudet.

Viimeisessä luvussa vertailtiin eri prosessoreita toisiinsa. Tässä luvussa tulee hyvin esille se kuinka ARMv6- ja ARMv7-arkkitehtuurit eroavat toisistaan. ARMv6-arkkitehtuurin prosessorit ovat ominaisuuksiltaan ja lisälaitteiltaan vähäisempiä mutta samalla halvempiäkin. ARMv7-perheen suorittimissa on taas paljon valinnaisia ominaisuuksia, joita valmistajat voivat joko tuoda tai poistaa prosessorista.

Juuri tämä valinnanvara eri prosessorien ja ominaisuuksien välillä on nostanut ARM-arkkitehtuurin suorittimet suosioon valmistajien keskuudessa.

Arkkitehtuurien ylöspäin yhteensopivuuden takia niillä on helppo aloittaa tuotekehitys halvemmista prosessoreista, minkä jälkeen voidaan siirtyä helposti kalliimpiin tuotteisiin tarvittaessa.

Mielestäni onnistuin tuomaan kaikki haluamani tiedot työn sisältöön, vaikka tarjolla olevan kirjallisuuden määrä oli todella suuri. Työssä oli juuri suurin pyökerätä kaikki tieto ja päättää mitä haluaisin pitää siinä. Aiheesta voisi saada aikaiseksi monta sataa sivua tekstiä, ellei tiivistä tekstiä halutulle tasolle.

Selvitystyö toi minulle uuden näkemyksen ARM-suorittimista ja Cortex-M-perheen mikrokontrollereista

LÄHTEET

Atack, C. & van Someren, A. 1993. The ARM RISC Chip: A Programmers Guide. Boston: Addison-Wesley.

Yiu, J. 2010. The definitive guide to the ARM Cortex-M3. Toinen painos. Amsterdam: Elsevier.

Yiu, J. 2011. The definitive guide to the ARM Cortex-M0. Amsterdam: Elsevier.

ARM Holdings plc. 2009. Cortex-M0 Devices Generic User Guide [viitattu 9.10.2013]. Saatavissa:
<http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dui0497a/index.html>

ARM Holdings plc. 2010a. Cortex-M3 Devices Generic User Guide [viitattu 9.10.2013]. Saatavissa:
<http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dui0552a/index.html>

ARM Holdings plc. 2010b. Cortex-M4 Devices Generic User Guide [viitattu 9.10.2013]. Saatavissa:
<http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dui0553a/index.html>

ARM Holdings plc. 2012. Cortex-M0+ Devices Generic User Guide [viitattu 9.10.2013]. Saatavissa:
<http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dui0662b/index.html>

ARM Holdings plc. 2013. Cortex-M Series [viitattu 8.10.2013]. Saatavissa:
<http://www.arm.com/products/processors/cortex-m/index.php>

Wikipedia. 2013. ARM Cortex-M [viitattu 8.10.2013]. Saatavissa:
http://en.wikipedia.org/wiki/ARM_Cortex-M