

DAFTAR ISI

HALAMAN PENGESAHAN.....	iii
PERNYATAAN.....	iv
KATA PENGANTAR	v
DAFTAR ISI.....	vii
DAFTAR TABEL.....	xi
INTISARI.....	xii
ABSTRACT.....	xiii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	2
1.4. Tujuan Penelitian.....	3
1.5. Manfaat Penelitian.....	3
1.6. Metodologi Penelitian	3
1.7. Sistematika Penelitian	4
BAB II TINJAUAN PUSTAKA.....	6
BAB III LANDASAN TEORI.....	8
3.1. Magnetic, Angular Rate, Gravity (MARG) sensor	8
3.1.1. Magnetometer	8
3.1.2. Girooskop.....	9
3.1.3. Akselerometer	9
3.2. Field-Programmable Gate Array (FPGA)	10
3.3. High Level Synthesis.....	11
3.4. Representasi <i>Quaternion</i>	12
3.5. Algoritme <i>Madgwick</i>	13
BAB IV PERANCANGAN SISTEM.....	15
4.1 Analisis Sistem	15
4.2 Perancangan Sistem.....	18
4.2.1 Perancangan koprosesor.....	18
4.2.2 Konfigurasi perangkat keras sistem	24
4.2.3 Perancangan perangkat lunak.....	25
4.3 Rencana pengujian.....	28
4.3.1 Pengukuran sumber daya	28

4.3.2	Pengujian akurasi	28
4.3.3	Pengukuran waktu latensi	29
BAB V IMPLEMENTASI		30
5.1	Implementasi rancangan koprosesor	30
5.1.1	Implementasi rancangan algoritme dalam C/C++	30
5.1.2	Simulasi C/C++ dan verifikasi koprosesor	39
5.1.3	High Level Synthesis	40
5.1.4	C/RTL Cosimulation	40
5.1.5	Ekspor	41
5.2	Implementasi konfigurasi perangkat keras	41
5.3	Implementasi perancangan perangkat lunak	42
BAB VI HASIL DAN PEMBAHASAN		44
6.1.	Hasil pengukuran sumber daya	44
6.1.1.	Sumber daya koprosesor	44
6.1.2.	Sumber daya sistem	47
6.2.	Hasil pengujian akurasi	47
6.2.1.	Hasil simulasi C/C++	47
6.2.2.	Hasil luaran perhitungan koprosesor	49
6.2.3.	Perbandingan data hasil perhitungan RMSE	55
6.3.	Hasil pengukuran waktu latensi	60
BAB VII KESIMPULAN		63
7.1.	Kesimpulan	63
7.2.	Saran	63
DAFTAR PUSTAKA		64
Lampiran 1 Data rekaman sensor MARG		66
Lampiran 2 Daftar <i>port</i> antarmuka koprosesor		67
Lampiran 3 Kode <i>testbench</i> dalam bahasa C/C++		68
Lampiran 4 grafik <i>golden reference</i> dalam representasi <i>quaternion</i> dengan sensor MARG		69
Lampiran 5 grafik <i>golden reference</i> dalam representasi <i>euler</i> dengan sensor MARG		70
Lampiran 6 grafik <i>golden reference</i> dalam representasi <i>quaternion</i> dengan sensor IMU		71
Lampiran 7 grafik <i>euler</i> dalam representasi <i>euler</i> dengan sensor IMU		72

DAFTAR GAMBAR

Gambar 3.1 Jenis-jenis magnetometer berdasarkan teknologinya (Lenz & Edelstein, 2006)	8
Gambar 3.2 Ruang vektor representasi giroskop (Starlino, 2009)	9
Gambar 3.3 Akselerometer normal, (b) Akselerometer kondisi + X, (c) Akselerometer kondisi -Z (d) Akselerometer berotasi sumbu Y (Starlino, 2009)	10
Gambar 3.4 Ruang vektor representasi akselerometer (Starlino, 2009)	10
Gambar 3.5 Arsitektur FPGA Xilinx (Brown, et al., 1992)	11
Gambar 3.6 <i>High Level Synthesis</i> dan logic synthesis (Crockett, et al., 2014)	12
Gambar 3.7 Orientasi dari <i>frame</i> B akibat rotasi yang segaris <i>frame</i> A dengan sudut θ sekitar sumbu Ar (Madgwick, 2010)	13
Gambar 3.8 diagram blok implementasi perkiraan orientasi algoritme lengkap untuk IMU (Madgwick, et al., 2011)	14
Gambar 3.9 diagram blok implementasi perkiraan orientasi algoritme lengkap untuk sensor MARG (Madgwick, et al., 2011)	14
Gambar 4.1 Blok diagram sistem	15
Gambar 4.2 Diagram alir perancangan sistem	17
Gambar 4.3 Diagram alir <i>Vivado HLS</i> (Crockett, et al., 2014)	19
Gambar 4.4 Pembagian algoritme <i>Madgwick</i> untuk sensor IMU	20
Gambar 4.5 Pembagian algoritme <i>Madgwick</i> untuk sensor MARG	20
Gambar 4.6 Penggabungan algoritme <i>Madgwick</i> untuk sensor IMU dan sensor MARG	21
Gambar 4.7 Rancangan program utama	26
Gambar 4.8 Rancangan program ISR <i>AXI timer 1</i>	27
Gambar 4.9 Rancangan program ISR koprosesor	27
Gambar 4.10 Rancangan program <i>ISR private timer</i>	27
Gambar 5.1 Sub fungsi normalisasi	31
Gambar 5.2 Sub fungsi perkalian <i>quaternion</i>	32
Gambar 5.3 Sub fungsi perkalian konstanta dengan vektor	32
Gambar 5.4 Sub fungsi penjumlahan dua vektor	33
Gambar 5.5 Implementasi deklarasi parameter fungsi algoritme <i>Madgwick</i> dalam bahasa C/C++	34
Gambar 5.6 Implementasi deklarasi variabel fungsi algoritme <i>Madgwick</i> dalam bahasa C/C++	34
Gambar 5.7 Implementasi orientasi dari kecepatan sudut dalam bahasa C/C++ ..	35
Gambar 5.8 Implementasi pengecekan akselerometer dalam bahasa C/C++	35
Gambar 5.9 Implementasi orientasi dari medan homogen sensor IMU dalam bahasa C/C++	36
Gambar 5.10 Implementasi pengecekan magnetometer dalam bahasa C/C++	36
Gambar 5.11 Implementasi orientasi dari medan homogen sensor MARG dalam Bahasa C/C++	37
Gambar 5.12 Implementasi proses fusi 1 dalam Bahasa C/C++	38
Gambar 5.13 Implementasi proses fusi 2 dalam Bahasa C/C++	39
Gambar 5.14 Implementasi perhitungan sudut <i>euler</i> dalam bahasa C/C++	39

Gambar 5.15 Proses verifikasi dan C/RTL <i>cosimulation</i> pada Vivado HLS (Crockett, et al., 2014)	40
Gambar 5.16 Hasil C/RTL <i>Cosimulation</i> desain RTL koprosesor AHRS	41
Gambar 5.17 <i>Report</i> berkas hasil proses <i>logic syhnthesis</i> dan <i>implementation</i>	42
Gambar 5.18 Konfigurasi perangkat keras.....	43
Gambar 6.1 Ilustrasi <i>trade off</i> antara optimasi sumber daya dan <i>throughput</i> (Crockett, et al., 2014)	44
Gambar 6.3 Grafik perbandingan <i>quaternion</i> hasil perhitungan algoritme dengan sensor IMU dalam C/C++ dan <i>golden reference</i>	50
Gambar 6.4 Grafik perbandingan <i>euler</i> hasil perhitungan algoritme dengan sensor IMU dalam C/C++ dan <i>golden reference</i>	51
Gambar 6.5 Grafik perbandingan <i>quaternion</i> hasil perhitungan algoritme dengan sensor MARG dalam C/C++ dan <i>golden reference</i>	52
Gambar 6.6 Grafik perbandingan <i>quaternion</i> hasil perhitungan algoritme dalam C/C++ dan <i>golden reference</i>	53
Gambar 6.7 Grafik perbandingan hasil luaran koprosesor dan hasil luaran koprosesor dengan sensor IMU dalam representasi <i>quaternion</i>	56
Gambar 6.8 Grafik perbandingan hasil luaran koprosesor dan hasil luaran koprosesor dengan sensor IMU dalam representasi <i>euler</i>	57
Gambar 6.9 Grafik perbandingan hasil luaran koprosesor dan hasil luaran koprosesor dengan sensor MARG dalam representasi <i>quaternion</i>	58
Gambar 6.10 Grafik perbandingan hasil luaran koprosesor dan hasil luaran koprosesor dengan sensor MARG dalam representasi <i>euler</i>	59
Gambar 6.11 diagram koneksi sinyal interupsi antara koprosesor AHRS	60
Gambar 6.12 <i>Cuplikan hasil C/RTL cosimulation</i>	61

DAFTAR TABEL

Tabel 2.1 Hubungan penelitian ini dengan penelitian sebelumnya.....	6
Tabel 4.1 Rencana pengujian	28
Tabel 6.1 Laporan perkiraan penggunaan sumber daya koprosesor dari <i>Vivado HLS report</i>	45
Tabel 6.2 Perkiraan rincian penggunaan sumber daya <i>Instance</i> dari <i>Vivado HLS report</i>	45
Tabel 6.3 Penggunaan sumber daya koprosesor AHRS.....	46
Tabel 6.4 Penggunaan sumber daya FPGA pada sistem.....	46
Tabel 6.5 Hasil perhitungan RMSE <i>quaternion</i> dari algoritme dengan sensor IMU dalam C/C++ dan <i>golden reference</i>	48
Tabel 6.6 Hasil perhitungan RMSE <i>euler</i> dari algoritme dengan sensor IMU dalam C/C++ dan <i>golden reference</i>	48
Tabel 6.7 Hasil perhitungan RMSE <i>quaternion</i> dari algoritme dengan sensor MARG dalam C/C++ dan <i>golden reference</i>	49
Tabel 6.8 Hasil perhitungan RMSE <i>euler</i> dari algoritme dengan sensor MARG dalam C/C++ dan <i>golden reference</i>	49
Tabel 6.9 Hasil perhitungan RMSE <i>quaternion</i> hasil luaran koprosesor dengan sensor IMU dan <i>golden reference</i>	54
Tabel 6.10 Hasil perhitungan RMSE <i>euler</i> hasil luaran koprosesor dengan sensor IMU dan <i>golden reference</i>	54
Tabel 6.11 Hasil perhitungan RMSE <i>quaternion</i> dari algoritme dengan sensor MARG dalam C/C++ dan hasil luaran koprosesor.....	54
Tabel 6.12 Hasil perhitungan RMSE <i>quaternion</i> dari algoritme dengan sensor MARG dalam C/C++ dan hasil luaran koprosesor.....	55
Tabel 6.13 Perbandingan data RMSE luaran koprosesor dengan <i>golden reference</i> dan simulasi C/C++ dengan <i>golden reference</i> dalam representasi <i>quaternion</i>	55
Tabel 6.14 Perbandingan data RMSE luaran koprosesor dengan <i>golden reference</i> dan simulasi C/C++ dengan <i>golden reference</i> dalam representasi <i>euler</i>	55