Computer Engineering and Applications Vol. 6, No. 1, February 2017

Priority based computation: "An Initial study result of paradigm shift on real time computation"

Sukemi, Riyanto

Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya
Technology and Innovation Centers Division (PUSTEKIN), PT. LEN INDONESIA
sukemi@ilkom.unsri.ac.id
riyanto@len.co.id

ABSTRACT

This research is purposed to increase computer function into a time driven to support real time system. This purposed would the processor can work according to determined time variable and can work optimally in a certained deadline. The first approachment to design some of processor that has a different bit space 64, 32, 16 and 8 bits. Each processor will be separated by selector/arbiter priority of a task. In addition, the design of the above processors are designed as a counter with varying levels of accuracy (variable precision computing). The selection is also done by using statistical control in the task are observed by the appearance of controller mounted on the front of the architecture bit space the second approach above. The last approach to 'add' certainty in the form of interval arithmetic precision cutting task that can be the upper bound and lower bound of the area (bounds).

These four approachment can be structured orthogonally into a processor/several processors by introducing a new classifier that serves as a selector or a task arbiter. The results of the four approaches to prove that the processor is prepared by incorporating a variable bitspace adder selectors can provide optimality of 0.43%.

Keywords: time driven, selector, variable precision computing, interval arithmetic.

1. INTRODUCTION

A system which drafted to improve reliability to rapid change (eg, changes in environmental conditions) that should give the right answers at the right time (timing correctness). In general, the concept is known as real time systems. Accuracy or delay of answers influences diverse characters that appear in the results of a computational process. Timeliness of an out answer affected by a range of response time is different for each system, depending on the demands of the environment in which the system works, and the process varies from a very short time span to that long one. The delay time of a system (or subsystem) to produce the right answers potentially raises the risk that threatens the success of other systems. [1-2].

Various approaches have been tested to meet the deadline on the real-time calculation system. This diversity can be divided into two groups: the software group, for example, the level of the operating system that focuses on developing systems Rate monotonic scheduling, (Lehoeczky et al., 1989), Non-preemptive, (Jeffay et al., 1991; Parks, Lee, 1995), Earliest Deadline First (EDF), Stankovic et al., (1998), Increase Reward Service (IRIS), Dey et al., (1993), Imprecise computation, Liu et al., (1991) and the level of the hardware group that seeks

through the processor clock speed or by enlarging the size of the granulation operation with k-bit, Real time arithmetic Units, Mora-Mora et al., (2006).

A simple review on the two groups approach to software and hardware above do not provide the ultimate solution is real-time system, although it has been developed back in the new scheduling techniques and computing hardware architecture.

In addition, a clock accelerated processor aims to build a system that does fast calculations performed with the effort to minimize the average value of the deadline for a number of task in a process [3]. This is different from the purpose of the construction real-time system which seeks to meet the deadline of each task. However, the most important thing from a real-time system is predictability, not the speed that has a meaning very relative. Each quick calculation will greatly assist in the fulfillment of short business deadline, but a quick calculation predictability alone cannot guarantee an exact-time system.

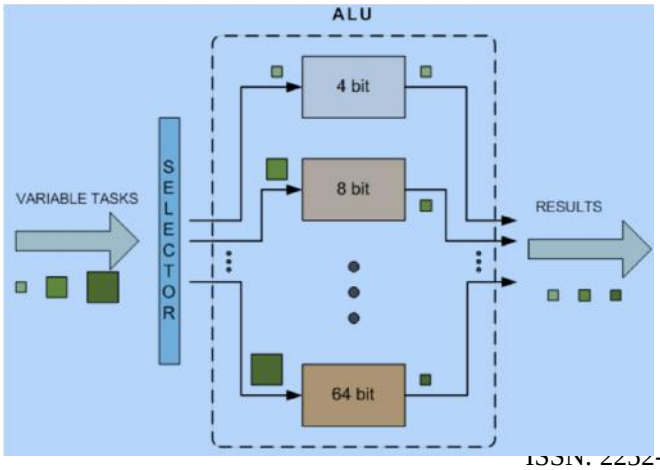
In fact, a real-time systems work correctly only if there are no timing failures. However, a timing failure will occur in varying amounts of time when the system is experiencing overloaded conditions (due to changes in environmental conditions) so that the system must be reset the task, a task that must be done to respond to these changes, although the real-time system with overload conditions cannot be avoided.

Degree of accuracy is also determined by the limited time available to achieve a real-time system. Therefore, the required limit or range of calculated results can provide answers in a decision-making process. Determination of the accuracy of the results of arithmetic calculations generally cannot be performed before the execution of the calculation is completed. The calculations may become vulnerable as a result of failure if the time available is not sufficient. If combined, timing and accuracy can determined degrees of precision of an operation such as arithmetic operations.

In addition to some things (timing, accuracy and precision) required timing selection, accuracy and precision be an interesting thing to be set for the system 'still' meet to real-time-ness. Interestingly, until now, there is no mechanism that can play the selector function represented by the hardware or software assigned to it. Every selection would consider the parameters of time, accuracy and precision so that the real-time system achieves its target [1].

2. SYSTEM REQUIREMENTS

The content of arithmetic adder structured on a model designed to be placed in each variable bitspace is built in stages in Figure 1 [4].



ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

FIGURE 1. Block model of variable bitspace adder [4]

All arithmetic adder functions are built on a MatLab R12a software tool in the granulation choice of 4, 8, 16, 32, and 64 bits and each will be associated with a selector that is placed at the front arithmetic architecture.

This selector also functions as a bitspace selector (granulation unit terms) aims to better ensure the timely achievement of a new computing paradigm for a particular arithmetic computational process.

3. SELECTOR FUNCTION

Selector function as a regulator of the use of resources, in this case the variable bitspaces adder precision of 4, 8, 16, 32, and 64 bits in accordance with the type and size of the incoming task. This is a little different from the bus arbiter arbitration which sets resources on the same bit width, such as 32-bit or 64-bit only.

In addition to the task set selector type, incoming bits width is adjusted to the task before processing in variable bitspace adder. By using this component, the expected time of execution of a task or series of tasks can be predicted faster in real-time.

In addition to the task selector has the function of selecting a variable bitspace, also designed selector accuracy of which are built in functions to obtain a value adder arithmetic accuracy despite the computing process has not been completed as a whole.

4. DESIGN ARCHITECTURE

The design of the variable bitspace adder that nested built from LUT FPGA totaling 16 blocks with granulation bit width $k = 4$ with the initial assumption that the width of the 64 bit operand. Each LUT contains 2 pieces of full adder (FA) which will perform the sum functions with the result of the upper limit and lower limit, as shown in the block diagram in Figure 2.

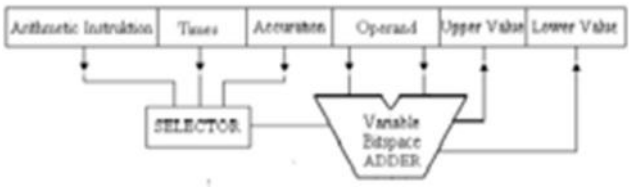


FIGURE 2. Block diagram of computational arithmetic adder variable bitspace

The difference in the computational calculations is primarily aimed at achieving the optimality of the time by installing a selector which functions as a sorting task or without selector.

The value of the accuracy-between can be a substitute accuracy when time constraints are not achieved or time restrictions by programmers who can be counted by (1) the following equation approach. Assuming that the maximum accuracy depends on the results of computing the value of the upper limit is currently underway computing. Accuracy-between is the accuracy of which is the width of the range between the upper limit value and lower limit value.

$$A_j(t)_{ars-antara} = \left[1 - \frac{\sum_{i=1}^{n} f(x)_{atas} - \sum_{i=1}^{n} f(x)_{bawah}}{\sum_{i=1}^{n} f(x)_{atas}} \right] * 100\% \quad (1)$$

The unit arithmetic of variable bitspace adder are designed with the following criteria, they can :

1. Can choose bitspace accordance with the bit width of task which will perform the computing process;
2. Can provide certainty when it generates a time-answer with an accuracy that could determine the level of its optimality;
3. Able to provide results-answer pair, in the form of an upper value and a lower value;
4. Able to produce answers with high accuracy since the beginning of the process of execution.

Realization of Figure 2, has been poured into the pattern of the arithmetic computing unit variable bitspace adder is designed with a width of 8 block and the bits of k = 4, shown in Figure 3.

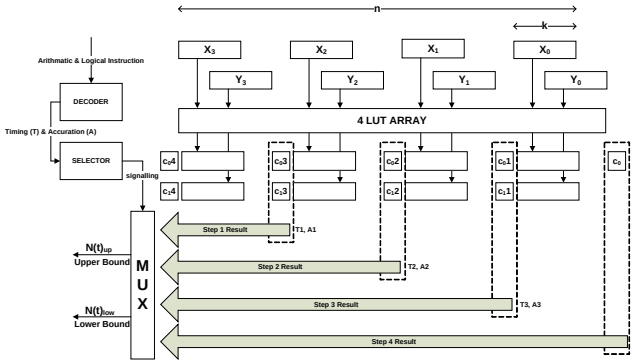


FIGURE 3. The pattern of 8 block computing of variable bitspace adder

The block of variable bitspace arithmetic adder in Figure 3 contains a decoder circuit, selector, *k*-operators, multiplexers and variable precision computing pattern. Each circuit serves to realize the the purpose of achieving the time optimality of variable precision accuracy if the minimum has been reached.

5. TESTING OF ARCHITECTURE DESIGN

The stages of testing and analysis of results of tests performed for each variable bitspace adder architectures (8 bit and 16 bit) that has been designed orthogonal.

The test results of arithmetic unit variable bitspace adder that will be reviewed in this paper only architectural patterns 8 bit and 16 bit are the result of the integration of the concept of some of the methods applied by means of simulation tools MatLab R12a. From the simulation results of this unit will look for optimal accuracy values at a desired time (time driven).

The following will be given a snippet of test results variable bitspace adder/sub on operand 8 bits of random data and a graph accuracy values obtained.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

The test results of variable bitspace adder arithmetic unit is the result of the integration of the concept of multiple methods through simulation released by MatLab simulator R12a. These test results (Figure 4) in the form of value upper limit and lower limit, maximum deviation, error and accuracy.
The testing of 8 bit single-operand (stand alone) with the amount of test data for 76.800 (equivalent to 1 layer (R or G or B) in the first frame), resulting in the achievement of high accuracy in the first cycle, which on average amounted to 89.09% (Table 1) with a value of error of 8% (the difference between the first cycle upper limit – lower limit first cycle).

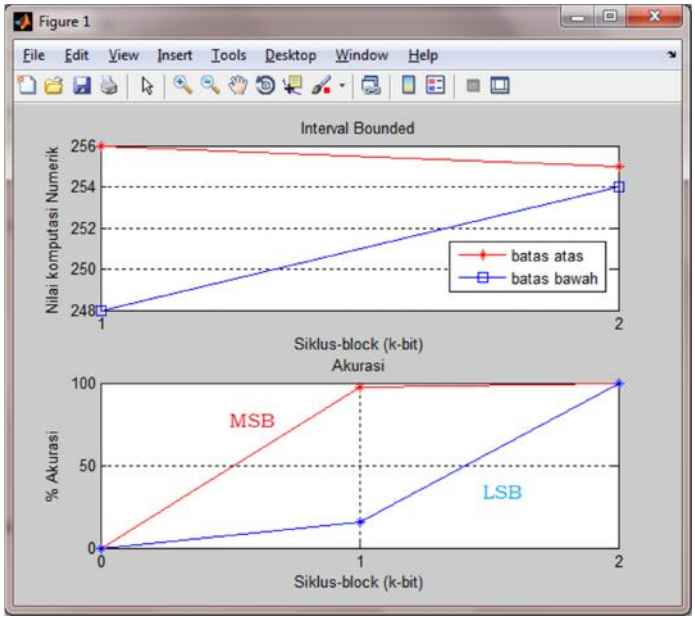


FIGURE 4. The resulting plot accuracy value in the variable bitspace adder/sub 8-bit single-operand, $k = 4$

The next test in the same bitspace but uses data that is multi-operand, as shown in Figure 5.

In testing the 8-bit multi-operand (8,16,32,64) with the amount of test data of 1000 data by the number of trials testing 10 times of the experiment resulted in the achievement of accuracy in cycles / bit operand 4th of 93.75% with a value of error of 16:25 %.

From some reference pictures, variable-bitspace architectural design shows that the accuracy of the results has been through the process of selecting selector / arbitrator with a wide selection of bitspace by 8 bits.

For bitspace 16, 32 and 64 bit multi input operand by the amount of data and the same experiment produced accuracy values both between and achievement of the desired level of accuracy as a programmer like in Figure 5.

Referring to Table 1. The testing on 100 single operand of random data, have shown that the accuracy of the results have been very high in the first cycle in the granulation $k = 4$ with an average value of 88.92%.

Value of accuracy in the first cycle of this MSB technique, when compared with the use of LSB technique that still exist in today's modern computer, has a difference of accuracy of 81.86%.

This value is derived from the difference in accuracy values in the first cycle of each technique, in which the technique of large LSB accuracy at the first cycle average on random data 100 by 7.05%.

TABLE 1.
Average accuracy value of variable-bitspace adder/sub 8-bit single operand

Data Random	Nilai Akurasi 8 bit *single operand*						Data Random	Nilai Akurasi 8 bit *single operand*					
	Langkah 1		Selisih	Langkah 2		Selisih		Langkah 1		Selisih	Langkah 2		Selisih
	MSB	LSB		MSB	LSB			MSB	LSB		MSB	LSB	
1	85.83691	0	85.83691	100	100	0	51	83.72093	0	83.72093	100	100	0
2	84.61538	1.234568	83.38082	100	100	0	52	92.01278	10.4	81.61278	100	100	0
3	87.27273	10.11236	77.16037	100	100	0	53	78.13953	12.08791	66.05162	100	100	0
4	93.63868	1.234568	92.40411	100	100	0	54	92.45283	0	92.45283	100	100	0
5	93.96825	11.11111	82.85714	100	100	0	55	64.36782	3.030303	61.33751	100	100	0
6	92.68293	0.689655	91.99327	100	100	0	56	93.47826	3.614458	89.8638	100	100	0
7	91.50327	11.11111	80.39216	100	100	0	57	95.68106	7.438017	88.24305	100	100	0
8	87.5	4.477612	83.02239	100	100	0	58	90.56604	0.884956	89.68108	100	100	0
9	91.42857	4.477612	86.95096	100	100	0	59	90.18182	5.882353	84.29947	100	100	0
10	81.81818	12.32877	69.48941	100	100	0	60	88.53755	0	88.53755	100	100	0
11	86.15385	3.759398	82.39445	100	100	0	61	85.43689	13.04348	72.39341	100	100	0
12	85.5814	15.78947	69.79192	100	100	0	62	93.02326	8.045977	84.97728	100	100	0
13	89.15663	7.246377	81.91025	100	100	0	63	88.88889	13.51351	75.37538	100	100	0
14	88.88889	9.090909	79.79798	100	100	0	64	86.82171	9.433962	77.38774	100	100	0
15	92.891	1.675978	91.21502	100	100	0	65	80.50314	4.477612	76.02553	100	100	0
16	89.27039	13.04348	76.22691	100	100	0	66	88.88889	0	88.88889	100	100	0
17	92.18107	6.976744	85.20433	100	100	0	67	84.21053	15.78947	68.42105	100	100	0
18	91.56627	2.702703	88.86356	100	100	0	68	88.25623	3.448276	84.80795	100	100	0
19	91.73333	4	87.73333	100	100	0	69	85.61873	7.692308	77.92642	100	100	0
20	83.66013	5.882353	77.77778	100	100	0	70	88.46154	17.94872	70.51282	100	100	0
21	94.28571	5.882353	88.40336	100	100	0	71	90.22556	6.796117	83.42945	100	100	0
22	83.33333	3.030303	80.30303	100	100	0	72	81.63265	18.98734	62.64531	100	100	0
23	89.30233	3.030303	86.27202	100	100	0	73	92.21902	8.860759	83.35826	100	100	0
24	87.84314	0.884956	86.95818	100	100	0	74	92.92929	0	92.92929	100	100	0
25	92.78351	9.433962	83.34954	100	100	0	75	88.53755	5.882353	82.6552	100	100	0
26	94.48819	8.280255	86.20793	100	100	0	76	88.58131	7.438017	81.1433	100	100	0
27	88.57143	0.775194	87.79623	100	100	0	77	95.62682	7.913669	87.71315	100	100	0
28	84.9162	18.98734	65.92886	100	100	0	78	89.71963	2.290076	87.42955	100	100	0
29	88.47926	5.882353	82.59691	100	100	0	79	82.24299	15.78947	66.45352	100	100	0
30	85.22337	9.677419	75.54595	100	100	0	80	92.10526	3.448276	88.65699	100	100	0
31	82.42424	9.859155	72.56509	100	100	0	81	87.5	13.04348	74.45652	100	100	0
32	90.10239	0	90.10239	100	100	0	82	89.65517	13.97849	75.67668	100	100	0
33	89.86301	2.702703	87.16031	100	100	0	83	88.63636	1.369863	87.2665	100	100	0
34	92.51701	7.438017	85.07899	100	100	0	84	92.87599	3.355705	89.52028	100	100	0
35	88.07339	5.882353	82.19104	100	100	0	85	94.29658	7.438017	86.85856	100	100	0
36	83.9779	21.31148	62.66643	100	100	0	86	90.14085	1.030928	89.10992	100	100	0
37	79.16667	15.78947	63.37719	100	100	0	87	93.99478	2.439024	91.55575	100	100	0
38	89.38547	16.88312	72.50236	100	100	0	88	93.61702	4.273504	89.34352	100	100	0
39	92.87599	8.280255	84.59573	100	100	0	89	86.2069	14.89362	71.31328	100	100	0
40	92.5	9.219858	83.28014	100	100	0	90	94.70405	9.859155	84.84489	100	100	0
41	84.70588	3.614458	81.09142	100	100	0	91	92.1466	0.621118	91.52548	100	100	0
42	86.38498	2.439024	83.94595	100	100	0	92	84.9162	1.234568	83.68163	100	100	0
43	91.6996	13.97849	77.72111	100	100	0	93	87.76371	7.692308	80.07141	100	100	0
44	89.44099	6.666667	82.77433	100	100	0	94	94.48819	9.433962	85.05423	100	100	0
45	90.06623	11.11111	78.95511	100	100	0	95	94.11765	1.234568	92.88308	100	100	0
46	93.45794	2.040816	91.41713	100	100	0	96	89.19861	7.438017	81.76059	100	100	0
47	86.33094	5.882353	80.44858	100	100	0	97	83.53414	13.04348	70.49066	100	100	0
48	95.4386	9.677419	85.76118	100	100	0	98	93.77289	12.72727	81.04562	100	100	0
49	85.87571	15.78947	70.08623	100	100	0	99	86.48649	0	86.48649	100	100	0
50	89.92248	4.273504	85.64898	100	100	0	100	95.77465	0.564972	95.20968	100	100	0

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)

All the results in Table 1, the testing method MSB when compared with the methods of calculation and testing LSB proved extremely significant difference, as shown in Figure 4 and in Figure 5.

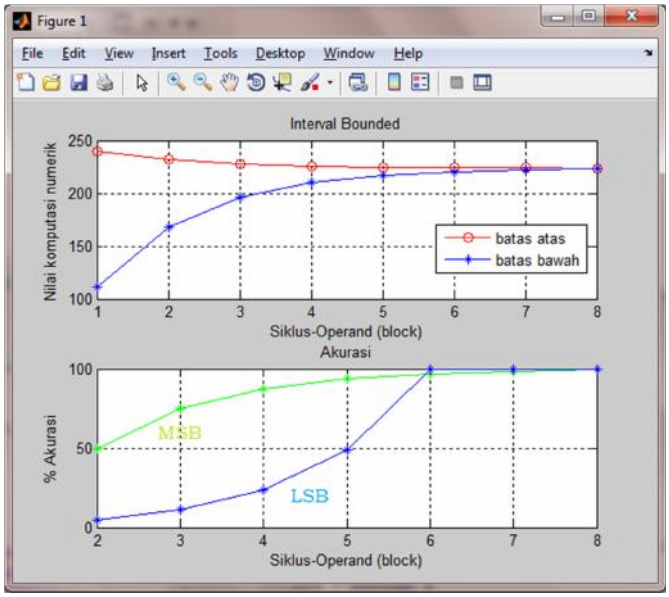


FIGURE 5. Plotting the results on the value of the variable bitspace accuracy adder/sub 8-bit multi-operand

The test results of real data (image 8 bits) with testing using random data, generated in the form of value upper limit and lower limit, maximum deviation, error and accuracy. The following will be given a snippet of test results bitspace variable adder/sub on operand 8 bit image data containing noise, as well as a graph accuracy values obtained. The test results are displayed in graphical user interface (GUI) in order to further facilitate the analysis of changes in the form of pictures and graphs generated.

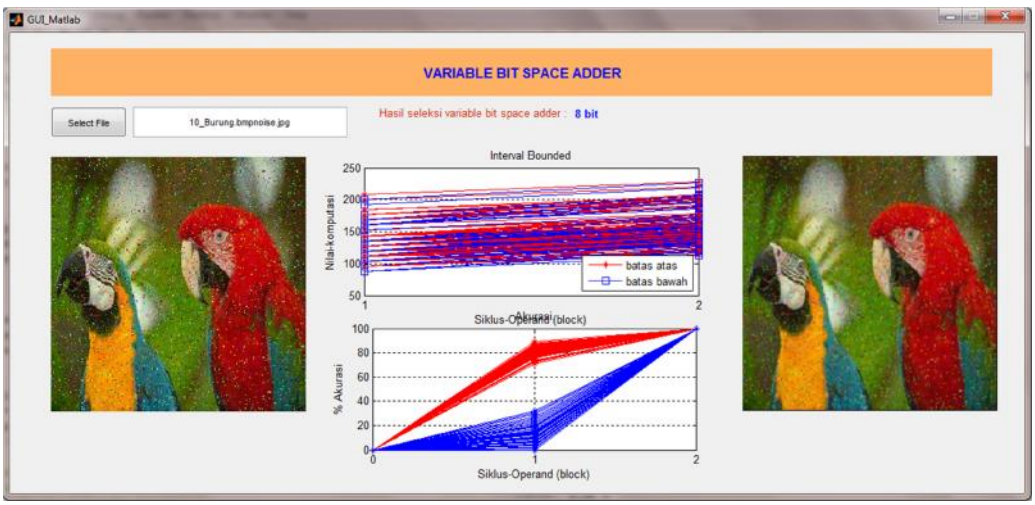


FIGURE 6. GUI variable bitspace k = 4 multioperand adder in 8 bit image data

In testing using real data in the form of 8-bit image above to produce accuracy is also high on the first cycle, which is an average of 88.90% for 1 frame (R or G or B) with a size of rows and columns each 10.

TABLE 2.
The simulation results on testing the accuracy of the average variable bitspace adder/sub 8-bit multi-operand.

Data random	Nilai akurasi 8 bit *multi-operand*						Data random	Nilai akurasi 8 bit *multi-operand*					
	Langkah 1		Selisih	Langkah 1		Selisih		Langkah 1		Selisih	Langkah 1		Selisih
	MSB	LSB		MSB	LSB			MSB	LSB		MSB	LSB	
1	88.13559	1.030928	87.10467	100	100	0	51	90.61489	4.273504	86.34138	100	100	0
2	74.66667	21.31148	53.35519	100	100	0	52	91.30435	1.234568	90.06978	100	100	0
3	84.55285	7.692308	76.86054	100	100	0	53	90.56604	10.4	80.16604	100	100	0
4	91.20521	9.677419	81.52779	100	100	0	54	91.92547	3.030303	88.89516	100	100	0
5	86.02151	14.66667	71.35484	100	100	0	55	91.20521	3.030303	88.17491	100	100	0
6	89.20863	7.438017	81.77062	100	100	0	56	88.88889	13.04348	75.84541	100	100	0
7	89.71963	5.084746	84.63488	100	100	0	57	89.76898	5.084746	84.68423	100	100	0
8	89.3401	9.090909	80.24919	100	100	0	58	82.92683	5.882353	77.04448	100	100	0
9	87.44939	13.97849	73.4709	100	100	0	59	91.80328	8.280255	83.52302	100	100	0
10	81.95122	7.246377	74.70484	100	100	0	60	91.69054	2.702703	88.98784	100	100	0
11	88.18898	3.448276	84.7407	100	100	0	61	91.58249	13.51351	78.06898	100	100	0
12	83.45324	11.92661	71.52663	100	100	0	62	90.60773	1.369863	89.23787	100	100	0
13	92.95775	5.882353	87.07539	100	100	0	63	89.24303	6.796117	82.44691	100	100	0
14	79.60199	0	79.60199	100	100	0	64	93.33333	4	89.33333	100	100	0
15	93.67089	7.246377	86.42451	100	100	0	65	88.88889	9.677419	79.21147	100	100	0
16	94.40994	2.290076	92.11986	100	100	0	66	95.2381	8.860759	86.37734	100	100	0
17	91.01796	4.477612	86.54035	100	100	0	67	89.82456	0.775194	89.04937	100	100	0
18	92.21184	5.882353	86.32949	100	100	0	68	91.01796	10.48951	80.52845	100	100	0
19	92.53012	2.762431	89.76769	100	100	0	69	91.00529	8.860759	82.14453	100	100	0
20	91.75627	9.677419	82.07885	100	100	0	70	92.71523	3.448276	89.26696	100	100	0
21	94.87871	2.439024	92.43968	100	100	0	71	81.70213	13.97849	67.72363	100	100	0
22	92.12121	9.219858	82.90135	100	100	0	72	94.62366	13.51351	81.11014	100	100	0
23	93.82716	7.246377	86.58078	100	100	0	73	89.49153	4.273504	85.21802	100	100	0
24	92.54499	8.280255	84.26473	100	100	0	74	92.06349	11.11111	80.95238	100	100	0
25	91.80328	3.448276	88.355	100	100	0	75	92.96636	1.538462	91.4279	100	100	0
26	93.48315	4.347826	89.13532	100	100	0	76	90.85873	9.859155	80.99957	100	100	0
27	90.51988	0	90.51988	100	100	0	77	93.25153	0.775194	92.47634	100	100	0
28	86.85714	3.030303	83.82684	100	100	0	78	89.53069	5.882353	83.64833	100	100	0
29	85.10638	15.78947	69.31691	100	100	0	79	90.85873	7.246377	83.61235	100	100	0
30	90.41096	5.084746	85.32621	100	100	0	80	82.28571	23.80952	58.47619	100	100	0
31	90.9699	10.4	80.5699	100	100	0	81	93.86667	7.692308	86.17436	100	100	0
32	93.99478	0.621118	93.37366	100	100	0	82	87.05882	6.569343	80.48948	100	100	0
33	84.84848	8.045977	76.80251	100	100	0	83	87.76371	3.030303	84.73341	100	100	0
34	88.88889	2.040816	86.84807	100	100	0	84	88.62275	0.775194	87.84756	100	100	0
35	89.85507	8.196721	81.65835	100	100	0	85	92.30769	11.81102	80.49667	100	100	0
36	90.43478	0	90.43478	100	100	0	86	87.33624	15.78947	71.54677	100	100	0
37	91.03448	11.11111	79.92337	100	100	0	87	94.70405	2.702703	92.00135	100	100	0
38	92.65537	7.246377	85.40899	100	100	0	88	80.95238	15.78947	65.16291	100	100	0
39	87.84314	13.97849	73.86464	100	100	0	89	91.95402	7.692308	84.26172	100	100	0
40	90.51988	0.689655	89.83022	100	100	0	90	91.00529	2.702703	88.30259	100	100	0
41	89.55224	10.28037	79.27186	100	100	0	91	86.66667	12.72727	73.93939	100	100	0
42	88.58131	6.666667	81.91465	100	100	0	92	93.92713	4.950495	88.97663	100	100	0
43	88.5906	4.477612	84.11299	100	100	0	93	90.06623	7.246377	82.81985	100	100	0
44	92.83276	0.775194	92.05757	100	100	0	94	91.42857	2.290076	89.1385	100	100	0
45	90.41096	11.11111	79.29985	100	100	0	95	88.88889	7.438017	81.45087	100	100	0
46	93.3687	4.635762	88.73294	100	100	0	96	91.16809	9.433962	81.73413	100	100	0
47	79.33884	5.882353	73.45649	100	100	0	97	93.87755	3.614458	90.26309	100	100	0
48	81.52866	3.030303	78.49836	100	100	0	98	81.5534	12.32877	69.22463	100	100	0
49	86.38498	13.04348	73.3415	100	100	0	99	89.57529	10.28037	79.29492	100	100	0
50	89.83957	4	85.83957	100	100	0	100	84.74576	2.439024	82.30674	100	100	0

Referring to Table 2, the test at 1000 random data on variables bitspace adder multi-operand with 100 attempts, has shown that the accuracy of the results have been very high in the first cycle in the granulation k = 4 with an average value of 89.35%. The first cycle accuracy value at this MSB technique when compared with the use of LSB technique that still exist in today's modern computer, has a difference of accuracy of 82.54%. This figure is derived from the difference in the value of the accuracy of cycle / first step of each technique, in which the technique of large LSB accuracy at an average first cycle in a random data in 1000 amounted to 6.99%.

Based on Table 1 and Table 2, between variables bitspace adder multi-operand with single-operand at the same bitspace (8 bit) has the distinction of accuracy, the accuracy of the variables bitspace adder multi-operand is higher than single-operand, which is 89.35% and 88.92%.

This has proved that the use of selectors within a variable bitspace adder/sub architecture can provide improved accuracy of 0.43%.

6. CONCLUSION

Apparently, the placement of the beginning of the task selector variable bitspace architecture adder/sub guarantee the accuracy of the optimality of time so that predictability can be determined prior to the completion of the computing process is completed. The optimal time can be evidenced by the increased accuracy of 0.43% with proven that more optimal if using selectors and collectors.

7. RESEARCH FUTURE

This paradigm shift has significantly missed and expected more significantly, if architecture variable bitspace adder will be tested on the data input derived from the actual environment (weather, medical analysis, data mining)

8. REFERENSI

[1]Sukemi, Ratna AAP., Sudibyo H, "Priority based computation: A Study of paradigm shift on real time computation," *Computational Intelegence and Cybernetics (Cyberneticscom), 2012 IEEE International Conference on Digital Object Identifier*, 12-14 Juli 2012, Bali, Indonesia. pp. 129-132, 2012.

[2]Stankovic, J.A., Spuri, M. dan Ramamritham, K., "Deadline cheduling for Real-Time Systems : EDF and Related Algorithms," Kluwer Academic Publishers (Kluwer International Series in Engineering and Computer Science), 1998.

[3]Sukemi, Ratna A.A.P, Sudibyo H, "Variable Bitspace of Variable Precision Processor," *Journal of Advanced in Information Technology*, pp. 65-70, 2014.

[4]Sukemi, Ratna A.A.P, Sudibyo H, "Development of an arithmetic unit bitspace architecture real-time for optimum performance," The *Book '3' part of dissertation,* examined on Mei 2016, Electrical Departemen, University of Indonesia, Depok, Indonesia, 2016.

[5]Sukemi, Ratna A.A.P, Sudibyo H, "Variable Bitspace Architecture with Variable Precision Processor," *The 2013 Annual International Conference on Advances*

Technology in Telecomunication, Broadcasting, and Satellite (TelSaTech 2013), 2-3 Agustus 2013, Jakarta, Indonesia. Adv. Sci. Lett. 20, pp. 531-533, 2014.

[6]Lehoczky, J., Sha, L. dan Ding, Y., "The Rate Monotonic Scheduling Algorithm: Exact Characterization And. Average Case Behavior," *Proceedings of Real Time Systems Symposium*, pp. 166-171, 1989.

[7]Parks, T.M. dan Lee, E.A., "Non-Preemptive Real-Time Scheduling of Dataflow Systems," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1995.

[8]Dey, J.K., Kurose, James, F. dan Towsley, D., "Efficient on-line processor scheduling for a class of IRIS (increasing reward withincreasing service) real-time tasks," *Proceedings of the 1993 ACM SIGMETRICS conference on Measurement and modeling of computer systems*, pp. 217-228, 1993.

[9]Liu, J.W.S, Shih, W.K., dan Lin, K.J., "Imprecise Computations," *Proceedings of the IEEE*. 82 (1), 1991.

[10]Mora-Mora, H., Mora-Pascual, J. dan Garcia-Chamizo, J.M., "Real-Time Arithmetic Unit," *Real-Time Systems*, 34 , pp. 53-79, 2006.

[11]Kerlooza, Y,K., Gondokaryono, Y,S., and Mulyana, A., "MSB-First Interval-Bounded Variable-Precision Real Time Arithmatic Unit," *ICT Journal*, ITB, 2010.

[12]Jeffay, K., Stanat, D.F., Martel dan C.U., "On Non-Preemptive Scheduling of Periodic and Sporadic Tasks," *Proceedings of the 12th Real-Time Systems Symposium*, 1991.

[13]wikipedia.org, "time driven programming", *http://en.wikipedia.org/wiki/Time-driven_programming*, 2014.

[14]Jensen ED., Locke CD., Tokuda H., "A-time-driven scheduling model for Real Time Operating Systems", *IEEE CH2220-2/b5/0000/0112$01.00*, 1985.

[15]Nielsen, asger Munk, Kornerup, Peter, "MSB-First Digital serial arithmatic," *Journal of Universal Computer science*, 1(7), 1995.

[16]David J. Kinnement, "Synchronization and Arbitration in Digital System," John Wiley., ISBN 978-0470-51082-7, 2007.

[17]Zadeh, Lotfi A, "Fuzzy logic, Neural Network, and softcomputing," *communication of the ACM*, 1994.

[18]Schulte, M.J.. "A Variable-Precision, Interval Arithmetic Processor," *PhD thesis University of Texas*, 1996.

[19]Hormigo, J., Villalba, J. dan Schulte, M.J., "A Hardware Algorithm for Variable-Precision Logarithm," *Proceedings of the IEEE International Conference on Application-Specific Systems, Architectures, and Processors*, 2000.

[20]Tinder, R.F., "Engineering Digital Design," 2. ed. San Diego, USA: Academic Press. — ISBN: 0-12-691295-5, 2000.

ISSN: 2252-4274 (Print)
ISSN: 2252-5459 (Online)