ETM

ENGINEERING & TECHNOLOGY MANAGEMENT

Title: Engineering Economic Analysis of an Electronic Product

Course:
Year: 1992
Author(s): N. Song

Report No: P92006

ETM OFFICE USE ONLY	
Report No.:	See Above
Type:	Student Project
Note:	This project is in the filing cabinet in the ETM department office.

Abstract: Cubical Calculus Machine II (CCMII) is a special hardware designed as a coprocessor for IBM personal computers (PC). This project is to conduct an economic engineering analysis of CCMII's market value.

ENGINEERING ECONOMIC ANALYSIS

OF AN ELECTRONIC PRODUCT

Ning Song

EMP-P9206

EMGT 535

Advanced Engineering Economic Analysis

ENGINEERING ECONOMIC ANALYSIS OF AN ELECTRONIC PRODUCT

Project Report

Submit to

Dr. Richard Deckro

Submitted by

Ning Song

Engineering Management Program
PSU

March 13, 1992

.. Introduction

Cubical Calculus Machine II (CCMII) is a special hardware designed as a co-processor for IBM personal computers (PC). This project is to conduct an economic engineering analysis of CCMII's market value.

Cubical Calculus includes a set of operations, mainly Sharp, Consensus, and Crosslink. These operations are found to be useful in logic synthesis, image processing, and automatic theorem proving, as well as other applications. Currently, these operations are executed by software. An on going project in the Electrical Engineering (EE) department at Portland State University (PSU) is to design a special hardware, which is a co-processor, to speedup these operations.

The hardware will be a board with some chips on it. The board will be plugged into one of the slots of an IBM/PC, either a 386 machine, or a 486 machine.

2. Speed analysis of CCMII applications

Since speed is the main objective of the product, we should first compare the speed difference in hardware and software. The software we used for testing is named EXORCISM, which is a software for AND-EXOR gate logic optimization. The software was written by students of PSU. There are standard benchmarks used by industry as well as universities for testing the results and the speeds of different software. By testing the same benchmarks, one software may generate better result than another software -- less gates for instance. Since the hardware designed is a co-processor, it does not change any operations or algorithms of the software. So, the results should be the same no matter the hardware is used or not. The only difference is the speed. A technical report was done analyzing the speed.[1] The following is the summary of the analysis:

1) <u>Timing analysis of a typical software</u>

Standard benchmarks are use to test the speed of the software. Time marks are set in the software to record the time. Table 1 shows the results of the testing. In table 1, adr4, log8, etc. are names of the benchmarks, Tt is the total time used for a benchmark, Ts is the time spent for sorting, and Tc is the time for cubical operations.

Table 1. Speed by using software[1]

Benchmark	Tt	Ts	Tc	Tc / Tt	Ts / Tt
adr4	10.650	1.550	8.717	0.818	0.146
log8	132.950	14.733	117.534	0.884	0.111
mlp4	96.383	13.983	81.417	0.845	0.145
nrm4	108.833	19.150	89.133	0.819	0.176
rot8	46.117	6.217	39.433	0.855	0.135
wgt8	159.717	30.950	127.700	0.800	0.194
rdm8	7.633	1.033	6.317	0.828	0.135
sqr8	224.150	26.117	196.850	0.878	0.117
Total	786.433	113.733	667.101	0.848	0.145

When using software, about 85% of the time is spent in cubical operations, and about 14.5% of the time is spend in sorting. Less than 1% of the time is spend for the rest of the operations.

2) <u>Timing analysis of the hardware</u>

The cubical operations can be divided into two categories: combinatorial operations and sequential operations. Let Tcc denote the time for a cubical operation using CCMII, and ns denote nano-seconds. For a combinatorial operation, Tcc = 500 ns. For a sequential operation, Tcc = 500 ns + Nr $\times$ 40 ns, where Nr indicates the number of resultant cubes.

When testing the software, not only the time, but also the numbers of cubical operations are recorded. Since we already know the time for each operation, the total time needed for cubical operations by hardware can be calculated.

'able 2. Computation time of CCMII[1]

	Nt	Ns	Nr	Ttcc (sec.)
.dr4	796872	1131	2318	0.399
.og8	17271961	1657	3434	8.636
ılp4	9791035	2045	4190	4.896
ırm4	12370615	1670	3432	6.185
·ot8	4793971	1279	2576	2.397
·gt8	14162130	2502	5069	7.081
·dm8	725251	745	1529	0.363
›qr8	27041610	2661	5355	13.521

In table 2, Nt is the total number of cubical operations executed in ·unning the respective benchmark. Among these Nt operations, the majority of :hem are combinatorial operations. The rest of the operations are sequential ›perations. Ns indicates the number of sequential operations performed by a ›enchmark. Nr is the number of resultant cubes generated in Ns operations. Ttcc .s the total time spent for cubical operations in that benchmark.

3) Time comparison of hardware and software

After we analyzed the time needed by the hardware and software, we :ompared the speed of software with the proposed hardware. Table 3 shows the :omparison between the hardware and software. Tt, Ts, Tc, and Ttcc have the same neaning as table 1 and table 2. Tc / Ttcc indicates how many times faster for :ubical operations the hardware was than the software. Tct is the total time ıeeded for a benchmark if a CCMII is used. In the case where a CCMII is used, :he cubical operations will be performed by the CCMII, but the sorting and the ›ther operations will be performed by software. In table 1, by average, about 1% ›f time is spent for the operations other than cubical and sorting (84.8% - 14.5% = 0.7 % ≈ 1%). So, Tct = Tcc + Ts + 0.01 × Tt.

'able 3. Comparison of hardware and software[1]

	Tt	Ts	Tc	Ttcc (sec.)	Tc / Ttcc	Tct	Tt / Tct
ıdr4	10.650	1.550	8.717	0.399	21.9	2.055	5.182
.og8	132.950	14.733	117.534	8.636	13.6	24.699	5.383
ılp4	96.383	13.983	81.417	4.896	16.6	19.843	4.857
ırm4	108.833	19.150	89.133	6.185	14.4	26.424	4.119
'ot8	46.117	6.217	39.433	2.397	16.5	9.075	5.082
‹gt8	159.717	30.950	127.700	7.081	18.0	39.628	4.030
'dm8	7.633	1.033	6.317	0.363	17.4	1.472	5.185
›qr8	224.150	26.117	196.850	13.521	14.6	41.880	5.352
'otal	786.433	113.733	667.101	43.478	15.3	165.075	4.764

In table 3, we can see that for cubical operations, the hardware will be 15 ;imes faster than the software. However, the total running time using the ıardware is only about 4.8 times faster than using the software.

3. Speeds and prices of general purpose processors

Special hardware has to compete with general purpose hardware. One of :he major advantages of using special hardware is their speed. However, new generations of general purpose hardware usually also have higher speed.

1) Technical forecasting

The Intel X86 family is the current dominant micro-processor in the PC narket. The following table shows the year the product entered the market, and the speed of the product. While the speed of the operations depend on many factors, the frequency is one of the most important factors.

Table 4. The trend of speed of Intel micro-processor

	Year	Speed
286	1987	10 MHz
386	1987	20 MHz
486	1990	33 MHz
586	coming	50 MHz

Source: investigate by the author

The software was tested on a Sparc II station, which is about 3 times faster than a 486 machine, and about 2 times faster than a 50 MHz machine. The comparison in previous section is between the CCMII and the software running on a Sparc II station. If the software runs on a 486 PC, the CCMII will be approximately 15 times faster than the software. And for a 50 MHz machine, the CCMII will about be 10 times faster than the software.

2) Price forecasting.

Currently, a 386 machine is about $2,000 and a 486 machine is about $3,000. By a straight line estimate, the coming 50 MHz machine would be about $4,000.

4. Cost analysis

1) Chip

Designing the chip is one of the major task of this project. Four chips are need for one CCMII. The cost is $400 for 4 chips. In the next step of development, if we decide to produce the CCMII in a large volume, the price will be much lower, about $20 for each chip.

2) Material & Parts

Besides the self-designed chips, other standard parts can be purchased from outside. The following is a list of standard parts required and their prices.

Table 5. List of components and prices[1]

Item	Type	Quantity	Unit Price	Total Price
D - Flip flop D - Flip flop Multiplexer Multiplexer 4 bit Latch	SN54LS273 SN54L74 SN54LS153 SN54LS157 SN54LS77	4 1 2 1 1	1 1 1 1 1	4 1 2 1 1
PLD PLD	GAL16V8 GAL22V10	3 2	2 5	6 10
Board Accessory		1	60	60 20
Total				105

3) Equipments

Two pieces of equipments are needed for this project. One is a PAL programmer which is used for writing the codes to PAL chips. The other key piece of equipment is a PC.

PAL Programmer $500*

PC/AT $1,500

4) Software

Current application software can not automatically interface with the CCMII. In order to take advantage of the CCMII, these software should be modified and re-compiled. In addition, we should write the software as part of the product package which links the application software and the CCMII. If we write this software in C, we need a C compiler. Another software required is CUPL, which prepares the files for PAL programmer.

---

* Electronic Engineering Times, Nov. 18, 1991, p. 143.

C++	\$350 (1)
CUPL	\$300
Software written in house	\$8,000 (This money can be saved if the job can be done as a student's project)

). Market Forecasting

Our potential customers will be the hardware companies in the area of logic optimization, and image processing. The product is specialized with a narrow market. By inquiring several of experts, the subjective estimate is that we may have 1000 customers with a decreasing exponential distribution.

Since our product is unique, there are no similar products currently in the market. We tried to look at some other products in the market to determine the price of our product. The following products are designed as a board plugged in PC with different functions.

1) 200 MHz Logic Analyzer (2) a board with software, \$799 - \$1,899

2) IEEE 488 Digital Interface Board (3) \$995

3) Modular VME Board (4) a computer graphic controller \$2,000

4) EISA single-board Controller (5) (bus-master control) \$1,845

5) 68HC11 PC-Based Emulator (6) \$2,590

---

(1) An educational package is cheaper than this price. Assume we use a commercial product.

(2) Electronic Engineering Times, Dec. 16, 1991, p. 79.

(3) Product Life, Jan. 6, 1992, p.

(4) and (5) same as above

(6) Electronic Engineering Times, Apr. 8, 1991, p. 62.

The suggested price for our product is $1,000. This price includes the hardware (the board with CCMII in it) and the software (link the application software and the CCMII board).

If the customers are not satisfy with such price, we can also sell it at $800 or $600. We think $600 will be very reasonable.

6. Benefit Cost Analysis

The costs are incurred in three phases: prototype, manufacturing, and marketing.

1) Prototype

In this phase, the costs are:

$400 for chips;

$105 for standard parts;

$2,000 for equipments;

$8,650 for software;

Total $11,155.

2) Manufacture costs

In this phase, no new equipment is needed. Because of the large volume, the cost for each chip is about $20. We expect 20% discount on standard parts. The costs are

$80,000 for chips ($20 × 4 × 1,000);

$84,000 for standard parts ($105 × 0.8 × 1,000);

Total $164,000.

3) Expected Income

Based on our estimation in section 5 (1,000 with decreasing exponential distribution), table 6 shows the probabilities of different volume of sales.

Table 6. Probabilities of sales

Volume of sales	Cumulative Probability	Probability
100	0.095162	0.095162
200	0.181269	0.086106
300	0.259181	0.077912
400	0.329679	0.070498
500	0.393469	0.063789
600	0.451188	0.057719
700	0.503414	0.052226
800	0.550671	0.047256
900	0.593430	0.042759
1000	0.632120	0.038690
1100	0.667128	0.035008
1200	0.698805	0.031676
1300	0.727468	0.028662
1400	0.753403	0.025934
1500	0.776869	0.023466
1600	0.798103	0.021233
1700	0.817316	0.019212
1800	0.834701	0.017384
1900	0.850431	0.015730
2000	0.864664	0.014233
more than 2000	1.000000	0.135335

The discount rates for parts and chips depend on the quantities we will order. Table 7 shows the costs per chip and the discount rates for parts corresponding to different quantities we may order. The prices are subject to change. The data in table 7 are estimated based on current available prices.

Table 7. Cost of chips and discount rate of parts

Number of boards will be produced	Cost per chip	discount rate for standard parts
50 -- 100	90	4%
101 -- 200	80	5%
201 -- 300	70	6%
301 -- 400	60	7%
401 -- 500	50	8%
501 -- 600	45	10%
601 -- 700	40	12%
701 -- 800	35	14%
801 -- 900	30	16%
901 -- 1000	25	18%
1001 and more	20	20%

Based on table 6 and table 7, we can calculate the expected income of the product. In the following table:

Revenue = Expected Value of Sales × price (assume $1,000);

Costs = Fixed costs + Variable Costs;

Fixed Costs = $11,155

Variable Costs = (Price of the chips for one board +

+ Price of the parts for one board) × quantity

Table 8. Expected Income

Probability (a)	Expected volume of Sales (b)	Revenue (c)	Costs (d)	Expected income [(c)-(d)] x (a)
0.095162	50	50,000	34,195	1,504
0.086106	150	150,000	74,118	6,534
0.077912	250	250,000	105,830	11,233
0.070498	350	350,000	129,333	15,557
0.063789	450	450,000	144,625	19,480
0.057719	550	550,000	162,130	22,387
0.052226	650	650,000	175,215	24,796
0.047256	750	750,000	183,880	26,753
0.042759	850	850,000	188,125	28,301
0.038690	950	950,000	187,950	29,484
0.035008	1,050	1,050,000	183,355	30,340
0.031676	1,150	1,150,000	199,755	30,101
0.028662	1,250	1,250,000	216,155	29,632
0.025934	1,350	1,350,000	232,555	28,981
0.023466	1,450	1,450,000	248,955	28,185
0.021233	1,550	1,550,000	265,355	27,278
0.019212	1,650	1,650,000	281,755	26,288
0.017384	1,750	1,750,000	298,155	25,240
0.015730	1,850	1,850,000	314,555	24,153
0.014233	1,950	1,950,000	330,955	23,044
0.135335	2,050	2,050,000	347,355	230,428
Expected income				689,301

In table 8, the expected income of the product is $689,301. This figure is calculated without considering the time value of the money. Next, we will consider the discount rate, and modify the calculation.

Initial costs incur at the beginning of the period. The cost of writing software incurs at the end of next 6 months. The manufacturing costs incur at the end of the month 6. We assume the revenue incurs continuously in the second 6 months. A 10% annual discount rate is assumed. In this case, the discount rate for one month is about 0.8%.

Based on above assumption, we have:

i) Initial costs is $3,155. NPV of initial costs = -$3,155

ii) At the end of the period from month 1 to month 6: software cost is

3,000 / 6 = $1,333

iii) NPV of the software is

1,333 × (P/A, 0.8%, 6) = 1,333 × 5.83552 = $7,781

In table 9, NPV of variable costs are calculated based on table 6. NPV of total costs = NPV of variable costs + $3,3155 + $7,781. NPV of expected income is calculated by (NPV of revenue - NPV of total cost) × probability.

Table 9. NPV of expected income

Probability (a)	NPV of Revenue (b)	NPV of Variable Costs (b)	NPV of Total Costs (d)	NPV of Expected Income [(b)-(d)] x (a)
0.095162	46,544	21,964	32,900	1,298
0.086106	139,633	60,023	70,959	5,913
0.077912	232,722	90,255	101,191	10,248
0.070498	325,810	112,660	123,596	14,256
0.063789	418,899	127,239	138,175	17,907
0.057719	511,987	143,927	154,863	20,613
0.052226	605,076	156,401	167,337	22,862
0.047256	698,165	164,661	175,597	24,695
0.042759	791,253	168,708	179,644	26,152
0.038690	884,342	168,541	179,477	27,271
0.035008	977,430	164,161	175,097	28,088
0.031676	1,070,518	179,795	190,731	27,869
0.028662	1,163,607	195,430	206,365	27,437
0.025934	1,256,696	211,064	222,000	26,835
0.023466	1,349,784	226,698	237,634	26,099
0.021233	1,442,873	242,333	253,268	25,260
0.019212	1,535,961	257,967	268,903	24,344
0.017384	1,629,050	273,601	284,537	23,374
0.015730	1,722,139	289,236	300,171	22,368
0.014233	1,815,227	304,870	315,806	21,342
0.135335	1,908,316	320,505	331,440	213,407
Expected income				637,637

By considering the time value, the expected income is $637,637, which is slightly smaller than the previous result. We plan that the product will be sold within one year. The time value is therefore a less critical consideration.

4) Marketing Expenses

At this time, we have not decided how to market our product. For this eason, marketing expenses are difficult to estimate. If we use advertising, the xpenses may be huge. On the other hand, if we do not use advertising, only a small processing fee is needed. In any event, we should keep in mind that the narketing expenses should be deducted from our expected income.

7. Break-even Analysis

Table 8 shows that even by selling 50 boards, we can earn profit. So, the break-even point is less than 50. Since the volume is small, we assume no discount for the chips and parts.

Given:

i) Fixed cost = \$3,155 (initial) + \$8,000 (software) = \$11,155

ii) Variable cost = \$100 × 4 (chips) + \$105 = \$505

Let Q indicates the number of boards sold, we have

i) $\$1,000 \times Q = \$505 \times Q + \$11,155$

ii) $Q = 11,155 / (1,000 - 505) = 22.53 \approx 23$

The break-even point is 23 boards.

The sales expenses are not considered in this analysis. If we assume \$20 for processing each order, then:

$$\$(1,000 - \$20) \times Q = \$505 \times Q + \$11,155$$

$$Q = 11,155 / (980 - 505) = 23.48 \approx 24$$

The break-even point is 24 boards.

8. Sensitivity analysis

Three parameters are important for this project: price, quantity, and marketing expenses.

1) <u>Price</u>: assume the range is from \$100 to \$1200, assume the quantity is 1,000.

Given:

i) Fixed cost = $11,155;

ii) Variable cost = 1,000 × 185 = 185,000

iii) Total cost = $185,000 + $11,155 = $196,155

If price = $100:

i) Revenue = $100 × 1000 = 100,000 when price = $100

ii) Benefits - costs = 100,000 - 196,155 = -$96,155

If Price = $1,200:

i) Revenue = $ 1,200 × 1000 = $1,200,000 when price = $1,200

ii) Benefits - costs = 1,200,000 - 196,155 = $1,003,845

Break-even point = 100 + 96,155 × (1,200 - 100) / (1,003,845 + 96,155)

= 100 + 96,155 × 1,100 / 1,100,000 = 100 + 96.155

= $196.155

The break-even point is about $200.

2) <u>Quantity</u>: Assume the range is from 0 to 2000, with a price of $1,000.

If the quantity sold is 0, the income is -$11,155

If the quantity sold is 2,000, income is

$1,000 × 2,000 - ($185 × 2,000 + $11,155) = $2,000,000 - $381,155

=$1,618,845

As we calculated in section 6, the break-even point is about 23 boards.

3) <u>Marketing Expenses</u>: Assume the range is from 1% to 40% of the price

If market expenses = 1% of the price:

$1,000 × 0.99 × 1,000 - ($185 × 1,000 + $11,155) = $990,000 - $196,155 =

= $793,845

If market expenses = 40% of the price:

$1,000 × 0.60 × 1,000 - ($185 × 1,000 + $11,155) = $600,000 - $196,155 =

= $403,845

9. Conclusions and Recommendations

1) The initial cost is small, the risk is not high

Because the PAL programmer, the CUPL, and the PC are available at the EE department, in the first phase the real cost is about $500. If we produce this product according to customer´s order, we have no risk of losing a lot of money.

2) We have not decided how to market this product. Using advertising is risky, because the expenses are high. One way I suggests is to co-operate with an application software producer. Our product would not be useful unless the customer uses the application software. We can share our profit with the application software producer, and sell the product along with the software.

3) According to the sensitivity analysis, the most important parameter is the number of boards we can sell. Since the potential users will be the software companies, not the private PC users, the quality of the product is more important than the price. If our product is proved to be very useful, then $1,000 is not a major consideration. Otherwise, reducing the price to $600 will not help sales very much.

4) Our product is only 5 times faster than the software running in a workstation. The main reason is that 15% of the time, the software is performing sorts. Even if the operations in the hardware can be 1,000 times faster than in the software, the total computation time can only be reduced by 15%, which is 6.67 times faster than the computation without special hardware. One method to solve this problem is to use another hardware called a content addressable memory chip (CAM). By using a CAM, instead of normal random access memory (RAM), the sorting time can be significantly reduced. The price and the effects of a

CAM should be further investigated however.

5) In the area of logic optimization, more users use workstations than PCs. Our current product is designed as a co-processor of a PC. With a small modification, the product can be extended to be the co-processor for a workstation (Sun station for instance). The market will then be much larger than the PC market.

6) Only one software is tested so far. We need test more software to compare with the hardware. The comparison with one software is not enough to convince the customers.

7) We need find more applications for our product. The more applications, the more customers we will have. If we find our product can speed up a specific application software, then we can try to co-operate with the software producer, and make our product useful for them.

8) Further market survey is needed. For each application software, we should know how many customers using that software. We can get this information from software companies.

9) Assembly costs and shipping have not been evaluated yet. We need estimate these costs and justify the results.

10) After we build the prototype, we can sell our design to a manufacture company. As a university project, it seems different to directly go to the market.

10. References

[1] Ning Song, "Timing analysis on CCMII", Technical report, EE department, PSU, March 1992.

[2] G. A. Fleischer, Engineering Economy, Wadsworth, Inc., 1984.

[3] R. R. Levary and N. E. Seitz, Quantitative Methods for Capital Budgeting, South-Western Publishing Co., 1990.