

Contents

1	Mailing List Archive	1
1.1	Friday 15th February Status report. (sent by Diarmuid)	1
1.2	Phase 1 Skeleton Code (sent by Edsko)	1
1.3	Notes on Visual C++ and minor code update (sent by Edsko)	8
1.4	Meetings (sent by Gerard)	9
1.5	RE: Meetings (sent by laura)	10
1.6	Updated feasibility report (sent by Diarmuid)	10
1.7	Updated feasibility report (sent by Diarmuid)	10
1.8	CVS and Qt (sent by Gerard)	10
1.9	Re: CVS and Qt (sent by Gerard)	11
1.10	Useful macro (sent by laura)	11
1.11	CCR Bit names (sent by Gerard)	12
1.12	Perforce Demo (sent by Edsko)	12
1.13	RE: Perforce Demo (sent by Gerard)	12
1.14	corrections to program (sent by laura)	12
1.15	Code update (sent by Edsko)	13
1.16	CCR (sent by Edsko)	13
1.17	2ba7: 01/03/2002 Status report (sent by Diarmuid)	13
1.18	Re: 2ba7: 01/03/2002 Status report (sent by Michael)	14
1.19	Setting CCR Flags - Official Motorola Definition (sent by Alan)	14
1.20	or, ori, small issue. (sent by Diarmuid)	14
1.21	RE: or, ori, small issue. (sent by Edsko)	15
1.22	RE: ori, RE: flags (sent by Edsko)	15
1.23	CCR (sent by Edsko)	15
1.24	Project Specification (sent by Alan)	16
1.25	RE: CCR (sent by laura)	16
1.26	RE: CCR (sent by Edsko)	16
1.27	Specification Report (sent by Edsko)	16
1.28	FWD: Little help from your friends (sent by laura)	17
1.29	FWD: Little help from your friends (sent by Brian)	17
1.30	2ba7: Status Friday 8th March (sent by Diarmuid)	17
1.31	Some issues (sent by Edsko)	17
1.32	Good Luck! (sent by laura)	18
1.33	RE: Some issues (sent by Annie)	18
1.34	RE: Proposed Work Plan (Vague) (sent by Annie)	18
1.35	RE: Some issues (sent by Edsko)	19
1.36	RE: Some issues (sent by Annie)	19
1.37	2ba7 project (sent by laura)	19
1.38	RE: 2ba7 project (sent by Edsko)	20
1.39	(no subject) (sent by laura)	20
1.40	2ba7 proj (sent by laura)	20
1.41	RE: 2ba7 proj (sent by Edsko)	20
1.42	gui files (sent by Annie)	21
1.43	RE: gui files (sent by Edsko)	21
1.44	Re: gui files (sent by Gerard)	21
1.45	Friday 4th April status. (sent by Diarmuid)	21
1.46	Compiler errors (relating include and lib files) (sent by Edsko)	22
1.47	The brand spanking new GUI (sent by Diarmuid)	23
1.48	Instruction-to-string translation (sent by Edsko)	23

1.49	NOTE: Code update (sent by Edsko)	25
1.50	Saved you'se some work again ;) (sent by Edsko)	26
1.51	How to test the T.. files (sent by Edsko)	26
1.52	Number format in T files (sent by Edsko)	27
1.53	Compiling for console mode (phase1) (sent by Edsko)	27
1.54	moc files and cvs (sent by Edsko)	28
1.55	Final demo ideas (sent by Edsko)	28
1.56	DEADLINE! DEADLINE! DEADLINE! (sent by Edsko)	29
1.57	Discussion of the product. (sent by Diarmuid)	29
1.58	Status report (sent by Edsko)	29
1.59	Some stats on the code. (sent by Diarmuid)	30
1.60	QT stuff (sent by Alan)	30
1.61	RE: QT stuff (sent by Edsko)	31
1.62	RE: QT stuff (sent by Edsko)	32
1.63	Code changes (sent by Edsko)	32
1.64	Small issue: 68sim.cpp (sent by Edsko)	32
1.65	suggestion for help stack. (sent by Diarmuid)	33
1.66	Code changes (2) (sent by Edsko)	33
1.67	Re: Final demo ideas (sent by Diarmuid)	33
1.68	RE: Final demo ideas (sent by Edsko)	34
1.69	Documentation? (sent by Edsko)	34
1.70	System stack (sent by Edsko)	34
1.71	RE: System stack (sent by Edsko)	35
1.72	Re: Documentation? (sent by Michael)	35
1.73	Re: Documentation? (sent by <osulldps@cs.tcd.ie>)	35
1.74	Meeting tomorrow! (sent by Diarmuid)	36
1.75	Friday 19th April update (sent by Diarmuid)	36
1.76	gcc warnings (sent by Edsko)	36
1.77	Bugs and bugfixes, also explanation of I/O (sent by Edsko)	37
1.78	Bugs and bufixes, and cool tutorial program (sent by Edsko)	38
1.79	Bugs and bugfixes, suggestion (sent by Edsko)	39
1.80	Help stack problems (sent by Edsko)	40
1.81	ADD and AND Issues (sent by Alan)	40
1.82	Mailing List Archive PDF (sent by devriese@tcd.ie)	41
1.83	Makefile and GCC warnings (sent by Edsko)	41
1.84	Makefile for Solaris (sent by Edsko)	41
1.85	GUI Update (sent by Gerard)	42
1.86	RE: bugfixes (sent by Annie)	43
1.87	RE: GUI Update (sent by Alan)	43
1.88	PDF of CVS log (sent by Edsko)	43
1.89	PDF of CVS log (sent by Edsko)	43
1.90	PDF of CVS log (sent by Edsko)	44
1.91	The final package (sent by Edsko)	44
1.92	Re: The final package (sent by Alan)	44
1.93	RE: (none) (sent by Edsko)	45
1.94	Re: The final package (sent by Diarmuid)	45
1.95	Hex vs. Decimal (sent by Edsko)	45
1.96	Deadlines, Generic stack reset() and GUI (sent by Edsko)	46
1.97	RE: Hex vs. Decimal (sent by Alan)	47
1.98	Friday 26th: Status report (sent by Diarmuid)	47
1.99	Stack operations (sent by Diarmuid)	48

1.100 Re: Stack operations (sent by Diarmuid)	48
1.101 Stack operations (sent by Diarmuid)	48
1.102 Stack operations. (sent by Diarmuid)	48
1.103 RE: Proposed CD covers (sent by Annie)	48
1.104 away till tuesday (sent by Annie)	49
1.105 68ksim Update (new makefile) (sent by Alan)	49
1.106 68ksim Update (new makefile) (sent by Alan)	49
1.107 Final deadline (sent by Edsko)	50
1.108 Status report (sent by Diarmuid)	50
1.109 CVS repository closed (sent by Edsko)	50
1.110 Final code changes/cleanups (sent by Edsko)	50
1.111 Re: Final code changes/cleanups (sent by Diarmuid)	51
1.112 Test mail (sent by Diarmuid)	51
1.113 Test mail 2 (sent by Diarmuid)	52
1.114 Test mail (sent by Diarmuid)	52
1.115 Test mail 3 (sent by Diarmuid)	52
1.116 Division of tasks (sent by Edsko)	52
1.117 user manual attached (sent by Annie)	52
1.118 CLOSING EMAIL (sent by Edsko)	52

1 Mailing List Archive

Converted from MBOX format by m2l, ©Edsko de Vries

1.1 Friday 15th February Status report. (sent by Diarmuid)

(Sent at Sat Feb 16 13:01:01 2002)

hello,

On Tuesday the 12th we met with Michael Manzke in Arch5 and covered briefly what he wanted from the project. The emphasis of this project seems to be as teaching tool for students to understand how the microprocessor works. The project will have to contain context sensitive help for what is occurring on the simulation, be graphical enough to be useful and also be able to read S-records from a file and interpret them so that we can provide a suitable environment for the user to understand what is occurring and if not they can receive some assist with the context help.

On Friday 15th we met and discussed our roles in the project along with specifics of the project and tools to making the project easier to manage. such jobs have been divided and we are currently working towards that end. Also the Feasibility Study has been written and has been sent on to the project supervisor for approval. Having received that it will be submitted to the secretaries office on Monday. Further details can be obtained from the Study when it becomes available.

Diarmuid Power
Customer Relations

1.2 Phase 1 Skeleton Code (sent by Edsko)

(Sent at Sat Feb 16 23:00:02 2002)

Dear project members,

I have been working hard today (from 8.00am to 10.30pm to be exact :) to fulfill my first task as lead programmer :-)) The attached file contains the .cpp and .h files for the skeleton code for phase 1.

```
=====  
Code Overview  
=====
```

The program starts with main() in 68sim.cpp. This does the following:

1. Initialize a memory device using the MemDevice.cpp class of 1 megabyte.
2. Initialize a CPU, connecting it to the memory device created in step 1. Later on, we could write a MemoryDecoder class which inherits from MemDevice, and attach the CPU to this decoder, but for the moment just the memory device will be enough.
3. It prints a welcome message
4. It checks the number of parameters given to the program on the command line. The program should be invoked with

```
68sim program initial_pc
```

Where program is the name of the program you want to load (.h68 file) and initial_pc is the initial value of the PC. Note that if you want to specify this in hex, you will need to use C++ hex style, ie:

```
68sim test1.h68 0x400
```

5. It then opens the input file, parses the S-records and loads them into memory (MemDevice has a method to load S-Records), and closes the file again.
6. It checks if the value for the initial PC is a valid number, and if so, it sets the PC on the CPU. It also sets the initial stack pointer (A7) to the end of memory.
7. It then steps through the program in memory, using the M68300::Step() method. More information on this method below, for now it will suffice to note that this method executes the instruction pointed to by the PC. It loops until the CPU is halted. Currently the CPU is halted by a \$F000 instruction, so make sure you end your test files using a

dc.w \$F000

statement.

8. When it finished running through the program, it prints out a message to confirm that the execution went fine, and then exits the program.

=====
Class details
=====

MemDevice.cpp, MemDevice.h

This class defines a memory device of any size. It provides methods to store/retrieve a byte, word or longword from/to memory, and has a method to interpret S-records. Currently it has no error checking, so accessing a longword from an odd memory address will behave as expected (instead of giving an Address Error), and trying to access a byte outside of the memory range will probably cause a Segmentation Fault (or crash in Windows), rather than giving a Bus Error.

M68300.cpp, M68300.h

This class defines the actual processor. It defines all the registers (d0..d7, a0..a7, pc, ccr), and has methods for each of the valid instructions (add, cmp, etc.) For information on the DATA_REGISTER and ADDRESS_REGISTER macros used, please see the C++ notes below.

The class assumes user mode rather than supervisor mode. This limits the amount of available instructions and registers, but since this program is meant as a learning tool, presumably for first years, this is an advantage (we're hiding complexity) rather than a disadvantage.

The Step() method referred to in main() retrieves the instruction pointed to by the PC, and invokes the associated method (ie. for an ADD instruction it will invoke the add() method). This method is fully functional, except for the EOR instruction which has an incorrect opcode bitmap in CPUREF.PDF, and I couldn't figure out the right values.

A few notes on naming conventions of the methods for the instructions. For each different opcode bitmap in CPUREF.PDF (which is by the way available from Manzke's 1BA3 website), there is a separate method. For some instruction this is very straight forward: the NBCD instruction has one associated opcode bitmap, and therefore one associated method: nbcd().

Some instructions, such as BSET, are actually two different instructions, dependant on the operands. In the case of BSET, there is a dynamic and static version. If this is the case, the instruction will have two different opcode bitmaps and therefore two associated methods, in this case bsetDyn() and bsetStat().

Other instructions, such as ASL and ASR, are actually just one instruction, with only one opcode bitmap. This instructions will only have one associated method, and the name of the method will be the names of the two instructions combined. In the case of ASL and ASR, this would normally be asl_r(), however, there is one complication. Even though ASL and ASR are actually the same instruction, this instruction has two different versions: one for register operations and one for memory operations (see CPUREF.PDF for details). Therefore there are two methods after all: asl_rMem() and asl_rReg().

Instructions denoted as Bcc or SETcc in CPUREF.PDF have associated method names bxx and setxx respectively. (I did this to avoid conflicts: bcc is actually a valid instruction - branch if carry flag it set.)

Step() works by masking out the 'reg', 'mode', etc. bits from the opcode bitmap, and then comparing the result with the opcode bitmap for the instruction. If equal, it invokes the associated method in M68300.

=====
What you need to do
=====

Even though the Step() method in the M68300 class is fully functional, all instruction methods (add(), sub(), etc.) are still to be completed - currently they only output their name (for testing purposes). The method definitions for the instructions are split up in three files:

- MInstrA-C.cpp, instructions starting with A..C
- MInstrD-N.cpp, instructions starting with D..N
- MInstrO-Z.cpp, instructions starting with O..Z

All of these methods should perform whatever they should do as specified in CPUREF.PDF - this is an important document! Please be as exact as possible when implementing these methods; it is important that these methods conform to the specification. ***THIS IS ESPECIALLY IMPORTANT WHEN UPDATING THE CCR***. The individual flags in the CCR can be addressed with ccr.z, ccr.v, etc. It is really important to set these flags when executing an instruction, because instructions such as BHI, SETLS, etc. depend on the flags.

The other important thing to remember is to update the PC. The main program will keep stepping through memory until the CPU is halted, and it will be halted upon encountering an \$F000 instruction. However, when you forget to update the PC after you have executed an instruction, the CPU will be executing the same instruction over and over again and will never stop. Usually you will have to add 2 to the PC (the basic instruction opcode bitmaps are 2 bytes), but more often than not an instruction will need additional operands, and if this is the case, you will have to add more bytes to the PC accordingly.

Note that in the methods you don't have to worry about the bits in the

opcode map that are defined to be 1 or 0 - these bits identify the instruction and are used by Step() - however, in the method itself you already know what instruction is being executed so the only thing you have to worry about is the 'regs', 'mode', 'dir', etc. bits.

Note also that the illegal() method is the only method which has already been implemented, that is, it updates the PC (adds 2 to it) and returns (it doesn't call the (non-existent) Illegal Instruction Exception Handler though). So therefore if you compile the following program

```
org $400

illegal
dc.w $F000
```

And run it through 68sim, you should get something like (example ran in Linux, on Windows the first line wouldn't appear):

```
-----
edsko@edsko: /doc/trinity/2ba7 > 68sim tests/test1.h68 0x400
68300 Simulation
(C) Trinity College Dublin, 2002
```

```
4AFC illegal
F000 Program termination okay
-----
```

So when you finish, say, the NEG instruction, you could compile a program

```
org $400

neg d0
dc.w $F000
```

And run it: (example in DOS)

```
-----
C:\68sim> 68sim mytest.h68 0x400
68300 Simulation
(C) Trinity College Dublin, 2002
```

```
4440 Negating d0, setting Z flag, blah blah
F000 Program termination okay
-----
```

```
=====
Distribution of tasks
=====
```

The instructions are split up into three files for two reasons: readability and to be able to split up the work over the different team

members.

File MInstrA-C.cpp is to be completed by Gerard and Alan
File MInstrD-N.cpp is to be completed by Diarmuid and Annie
File MInstrO-Z.cpp is to be completed by Laura and myself

Each files contains approximately the same number of instructions.

=====
Notes on compilation
=====

For the moment, this program is a console program, so if you want to work with Microsoft Visual C++, you have to create a Windows Console Application. Then, before you run the program, you have to set the command line options (see below) in the MC++ settings (otherwise the program will complain about having too few parameters).

In UNIX/Linux, the program can be compiled with

```
g++ -o 68sim *.cpp
```

=====
C++ Specific Notes
=====

A number of notes of C++ features used in the skeleton code.

1. Hex numbers

Hex numbers in C++ are defined as

0xnumber, eg. 0x10A

(rather than \$10A).

2. Macros

An example of a macro in C++ is

```
#define DATA_REGISTER( reg ) \  
union \  
{ \  
  unsigned char b; \  
  unsigned short int w; \  
  unsigned long l; \  
} reg\
```

This is basically a search-and-replace mechanism. If we use

```
DATA_REGISTER( d0 );
```

this will be replaced with

```
union
{
unsigned char b;
unsigned short int w;
unsigned long l;
} d0;
```

The backslashes in the macro definition are required for multi-line macros. For more information on unions, please see below.

3. #ifndef

Consider the following example

```
#ifndef M68300_H
#define M68300_H
```

```
blah blah blah
```

```
#endif
```

The compiler will only insert the text in between the `#ifndef` and `#endif` statements if the constant `M68300_H` (in this example) has been defined. Within the block, we actually define this constant (albeit empty). This is used to avoid including the same header file more than once: the header file starts with the `#ifndef HEADERNAME_H`; if this is the first time the header file is included, this constant will not yet be defined and therefore the file will be included - and while it is included, the `HEADERNAME_H` constant will be defined by the `#define` statement - therefore should the header file be included a second time, the constant will have been defined and the file will not be inserted again.

4. Struct and bitmaps

A struct in C++ is a class without methods and only public attributes. However, structs have one advantage of classes, you can specify bitmaps. Example:

```
struct
{
char HB;
char reserved: 3;
char E: 1;
char N: 1;
char Z: 1;
char O: 1;
char C: 1;
```

```
} ccr;
```

The variable HB is a normal variable of type char. However, reserved is only 3 bits, and E, N, Z, O and C are only 1 bit in length. This makes it really easy to access the individual bits in a variable (and this is especially practical when trying to access the flags in the CCR).

5. Unions

A union in C++ is very similar to a struct, but with one major difference: all variables start at the same memory location. This is probably best explained with an example:

```
union
{
unsigned char b;
unsigned short int w;
unsigned long l;
} d0;
```

In this example, the variables b, w and l all starts at the same memory address; ie. if we change b

```
d0.b = 10;
```

we also change w and l - which is exactly what we want as d0.b, d0.w and d0.l are all part of the same register.

```
=====  
That was all for today ;-)  
=====
```

Have a look at the code, and try to understand what's happening. It is important that you understand everything in the skeleton code, as this will be the foundation for the instruction decoding engine.

If you have any questions, please send an email to the mailing list, rather than directly to me, as they might be (probably are) relevant to all of us, and I will reply to the mailing list as well. And please do not hesitate to ask questions: this will be the very core of our project, so this needs to be absolutely correct - if you are unsure about anything, do ask.

Happy programming :)

Edsko

(Skipping MIME encoded file . . . Ok)

1.3 Notes on Visual C++ and minor code update (sent by Edsko)

(Sent at Mon Feb 18 22:20:01 2002)

Hey all,

As Laura pointed out to me in the ICT labs, the code I sent out doesn't work in Visual C++. Of course I had developed and tested the program in Linux, and as I used standard C++ libraries only, I had assumed it would work in VC++ as well. But of course Microsoft had it's own ideas about "standard" C++ libraries...

Anyway, the attached ZIP-file contains the updated code which runs both in VC++ and in Linux (next time I'll test it for both... developing for two platforms turns out bugs quicker as well). The ZIP file now also contains a very simple .ASM file plus .H68 (and .LIS file) which can be used if you are able to compile the program correctly.

Here are detailed steps on how to get this to work in VC++:

1. Unpack the ZIPfile to a folder, say D:\DOC\68SIM
2. Start up VC++ and click File, New
3. Choose a Win32 Console Application, give the project the name "68sim" and set the correct path (say, D:\DOC) so that the location will be the same location and the location of the unzipped CPP and H files (say D:\DOC\68SIM) and click OK.
4. Select an empty project and click on finish, and okay
5. In fileview, add all .CPP files to "Source Files" (there should be 6 of them) and all .H files to "Header files" (2)
6. Double-click on 68sim.cpp to open that file
7. Go to Project, Settings
8. On the right hand side you can choose for General, Debug, C/C++, Link, etc. Choose for Debug
9. In program arguments, enter: test.h68 0x100
(these are the command line arguments to the program, and tell the program to load test.h68 into memory and start execution at 0x100, which is the C++ equivalent of \$100 in assembler), and click Ok
10. Click Build, Rebuild All - it should compile and link without any errors or warnings
11. If you run the program now, you should get

68300 Simulation
(C) Trinity College Dublin, 2002

4AFC illegal
F000 Program termination okay
Press any key to continue

12. If you did, congrats: the program is compiled and working :) If not, please let me know what's gone wrong.

(I'm trying these steps as I go and it works for me, on Windows98 and Microsoft Visual C++ 6.0)

Hope this solves the problems

Edsko

(Skipping MIME encoded file . . . Ok)

1.4 Meetings (sent by Gerard)

(Sent at Tue Feb 19 11:40:02 2002)

Hey all,

ok we'll be having a meeting on Friday after 2BA2 Lecture and every Friday thereafter. Any objections?

Gerard.

BTW: Chora, the CVS web interface is almost ready at <http://www.netsoc.tcd.ie/horde/chora/>

"As for any inner meaning or message, it has in the intention of the Author none." - J.R.R. Tolkien

1.5 RE: Meetings (sent by laura)

(Sent at Tue Feb 19 11:42:01 2002)

Nope, c yas there :)

none." - J.R.R. Tolkien

1.6 Updated feasibility report (sent by Diarmuid)

(Sent at Tue Feb 19 11:56:01 2002)

Hi,

This is the updated feasibility report, which now includes the task division, per programming pair, within phase 1, as discussed with Dr. O'Sullivan. Task divisions for future phases in the project will be defined and documented as we go.

In addition, the report mentions the use of the Qt graphics library as a cross-platform library to implement the graphical user interface.

Diarmuid Power
Customer Relations

1.7 Updated feasibility report (sent by Diarmuid)

(Sent at Tue Feb 19 15:07:02 2002)

Hi,

This is the updated feasibility report, which now includes the task division, per programming pair, within phase 1, as discussed with Dr. O'Sullivan. Task divisions for future phases in the project will be defined and documented as we go.

In addition, the report mentions the use of the Qt graphics library as a cross-platform library to implement the graphical user interface.

Diarmuid Power
Customer Relations

(Skipping MIME encoded file . . . Ok)

1.8 CVS and Qt (sent by Gerard)

(Sent at Thu Feb 21 11:28:02 2002)

Hey all,

ok Chora is not going to be happening. I got it to work alright=20 although I gave you the wrong address in my last email. Sorry about that. However, it was only a viewer, you could only see the files and download them, you couldn't actually update or add files. Unfortunately there doesn't appear to be any sort of web interface around which does this. This would leave us the only option of having to use the command line on matrix. However this isn't too bad, since all of us have matrix accounts and most of us are fairly proficient with UNIX. I'll see if I can write up a simple tutorial by tomorrow. If it turns out that some of us find CVS to be too complex, which I don't think that it should, then we could just make it the responsibility of one person in each pairing, thus solving the problem.

Next, Qt. Ok there is a file at <http://www.netsoc.tcd.ie/~gerard/qt.exe> which is the Qt library for windows. All you need to do is download this and run it, installing it on your computer. Then you need to change one setting inside Visual C++ to get it to work with VC++. You can also download it to one of the College computers installing it on the user D drive. I know that we won't be needing to use Qt for another couple of=20 weeks, however I would recommend that you at the very least have a=20 quick look at it, and would suggest that you look at one or two of the=20 tutorials.

Finally, just after reading through Edsko's emails there and realised=20 what exactly it was he had written last Saturday. Good work Edsko,=20

a very solid base to the project already.

Anyway, see you all at the meeting tomorrow.
Gerard.

1.9 Re: CVS and Qt (sent by Gerard)

(Sent at Thu Feb 21 13:29:01 2002)
whoops.... correct address is:
<http://www.netsoc.tcd.ie/~gerard/Qt.exe>

Sorry about that.
Thanks Annie.

Gerard.

1.10 Useful macro (sent by laura)

(Sent at Thu Feb 21 14:58:02 2002)
Here ya go!:)
#define GETBITS(addr, mask, shift) (memory->GetWord(addr) & mask) >>
shift

1.11 CCR Bit names (sent by Gerard)

(Sent at Thu Feb 21 18:10:02 2002)

I take it that the E and X bits are one and the same and that O and V =
are both the same as well.

Just slightly confused between differences in the manual and in your =
code Edsko.

Gerard.

1.12 Perforce Demo (sent by Edsko)

(Sent at Tue Feb 26 11:12:02 2002)
Hey,

As there are no web interfaces available for CVS, Mal installed an alternative system, called Perforce, which *does* have a nice and easy to use web interface. He offered to demo the system coming Friday in our group meeting - so all attend :)

Edsko

1.13 RE: Perforce Demo (sent by Gerard)

(Sent at Tue Feb 26 14:29:01 2002)

Mal has asked me to do the demo, so you'll have to suffer my fumbling! :-)

Gerard.

alternative

"As for any inner meaning or message, it has in the intention of the Author none." - J.R.R. Tolkien

1.14 corrections to program (sent by laura)

(Sent at Tue Feb 26 15:54:01 2002)

Hi ya,

I looked through the opcodes passed into the LINK and LINK_EXT methods and compared them to the opcodes in the new reference manual. There's only a few little corrections ;

-instructions 'cmp2', 'moveFromCcr', and 'chk2' can be omitted.

-i think the 1st parameter passed into the LINK macro for 'eor' should be ff00(not too sure tho). The opcodes for eor and cmp are right tho.

-There are some instructions included in our program that aren't included in the manual. These are;

bgnd, bkpt, tblsDataReg, tbluDataReg, tblsTableLookup and TblutableLookup.

-Likewise, there are insructions from the manual omitted from our program; bclr, rtr. (these ARE listed as mc68300 family)

-The 11th line down in the LINK_EXT macro list should read 'divuLong' as opposed to divsLong. Don't know if this was already changed or not.

Think that's it. Let me know if ya think i left anythin out. :)

1.15 Code update (sent by Edsko)

(Sent at Tue Feb 26 17:23:02 2002)

Hey,

For those who are interested, attached in a (intermediate) code update. The processor initialization will set the CCR and D/A registers to 0 when booting, and another instruction has been implemented: F001 which will prints out the registers (useful for debugging).

ie.

```
move d0,d1
dc.w F001
```

Edsko

(Skipping MIME encoded file . . . Ok)

1.16 CCR (sent by Edsko)

(Sent at Wed Feb 27 21:04:01 2002)

Hey,

I changed the definition of the CCR bits in my local copy of the M68300.h header file to unsigned chars (rather than signed chars).

If this is okay to everybody this change will propagate to the official version.

Edsko

1.17 2ba7: 01/03/2002 Status report (sent by Diarmuid)

(Sent at Fri Mar 1 15:33:01 2002)

hello,

Current state of the project puts us with the completion of Phase 1 on Monday. All teams are heavily stuck into instruction decoding and we hope to get as many instructions operating as possible.

CVS is up and running. The team was given an introduction by Gerard to CVS and how it will be used. Coding will continue over the weekend to get as much done in Phase 1 as possible.

Initial version of the next deliverable has been talked over and Alan is currently doing up the formatted version for approval.

The proposal was raised for one team to move onto context-sensitive help and the other two groups to proceed with Phase 2(GUI) of the project. This will allow for greater efficiency as regards adding the basic elements of the context sensitive help to the GUI as it is being developed. This will hopefully increase efficiency of the later stages of the project.

Diarmuid Power
Customer Relations

1.18 Re: 2ba7: 01/03/2002 Status report (sent by Michael)

(Sent at Sat Mar 2 15:40:01 2002)

Well done. I like the CVS bit.

Have a nice weekend.

Michael

Diarmuid Power wrote:

1.19 Setting CCR Flags - Official Motorola Definition (sent by Alan)

(Sent at Sat Mar 2 18:00:02 2002)

Hey all,

Thought you might like to know this if you didn't already... Pages 83 onwards in the Motorola Manual detail the conditions for setting the CCR flags `_per instruction_` for the M68k instruction set.

Might save a bit of head scratching :)

Alan

=====

adonnell@tcd.ie

1.20 or, ori, small issue. (sent by Diarmuid)

(Sent at Sat Mar 2 19:30:01 2002)

hey all,

Me and Annie were using the or and ori instruction implementation for our eor and eori instructions and noticed that when the `ccr.n` and `ccr.z` flags were being set, it was being done before the result had been stored somewhere.

D & A

The Dynamic Duo

1.21 RE: or, ori, small issue. (sent by Edsko)

(Sent at Sun Mar 3 11:07:01 2002)

That is because for SUB we needed both the result and the two individual operands to be able to set the flags - we made it a habit of storing the result to a temporary 'register' `DATA_REGISTER(result)`, and then setting the flags based upon that result register and the two operands.

It makes no difference of course (the flags won't be affected by the other operations, so we might as well set the result on our 'result register' as on the actual operand that `_will_` contain the result.

Edsko

1.22 RE: ori, RE: flags (sent by Edsko)

(Sent at Mon Mar 4 12:58:01 2002)

Yep, bug in ORI acknowledged, will fix.

Page in manual describing the official algorithms to set the CCR flags is at page 3-18 or 89 of the manual not 83. Thanks Alan for telling us though; I suggest we revise our code to make it conform these rules.

Edsko

1.23 CCR (sent by Edsko)

(Sent at Mon Mar 4 13:44:01 2002)

Hey,

Laura and I made a macro to set the CCR flags in the SUB(x) instructions:

```
/**
Set the C and V flags in the SUB instruction

S = source, D = destination, R = result, M = MSB8 — MSB16 — MSB32
*/
#define SUB_SETV( S, D, R, M ) \
ccr.v = ( ( ( S & M ) & ( D & M ) & ( R & M ) — ( S & M ) & ( D & M ) & ( R & M ) ) & M ) ? 1 : 0
#define SUB_SETC( S, D, R, M ) \
ccr.c = ( ( ( S & M ) & ( D & M ) — ( R & M ) & ( D & M ) — ( S & M ) & ( R & M ) ) & M ) ? 1 : 0
```

(The conditions are taken directly from the manual.) Example of usage of the macro:

```
SUB_SETV( d[ reg ].b, d[ eaReg ].b, result.b, MSB8 )
```

Where MSB8 is defined as 0x80 (MSB16 would be 0x8000 and MSB32 would be 0x80000000).

We thought this might be useful for the other groups as well (of course you will have to change the actual conditions).

Edsko

1.24 Project Specification (sent by Alan)

(Sent at Mon Mar 4 14:01:01 2002)

Hey all,

A draft of the Project Specification is attached. Needs a little work before submission, but it shouldn't take too long :)

Alan

```
=====  
adonnell@tcd.ie
```

(Skipping MIME encoded file . . . Ok)

1.25 RE: CCR (sent by laura)

(Sent at Mon Mar 4 14:04:01 2002)

Hey

Just wanna know if it's alright if I don't fix the sub carry flags 'til after end of this term. Is it? (Gerard, Alan I know you're waitin on it for 'cmp'
) Regards,
Laura :)

1.26 RE: CCR (sent by Edsko)

(Sent at Mon Mar 4 19:19:01 2002)

Yeah, that'll be fine (you and me are the only ones continuing on the instruction set anyway)

Edsko

1.27 Specification Report (sent by Edsko)

(Sent at Tue Mar 5 13:14:02 2002)

Hey all,

Specification report which has been submitted to Michael Manzke for approval can be downloaded from

<http://matrix.netsoc.tcd.ie/~edsko/68sim/spec.doc>

Edsko

1.28 FWD: Little help from your friends (sent by laura)

(Sent at Tue Mar 5 19:05:02 2002)

Get your FREE download of MSN Explorer at <http://explorer.msn.com/intl.asp>.

— Forwarded Message —

1.29 FWD: Little help from your friends (sent by Brian)
1.30 2ba7: Status Friday 8th March (sent by Diarmuid)

(Sent at Fri Mar 8 13:47:02 2002)

hello,

As of the end of term, we stand roughly 80
The GUI development is begining soon. Initially we need to familiarise ourselves with QT(running through one of the demonstrations for it) and

in parallel come up with the skeleton framework for the GUI. From there the work can be divided and dealt with accordingly.

Diarmuid Power,
Customer Relations.

1.31 Some issues (sent by Edsko)

(Sent at Fri Mar 8 16:01:02 2002)

Hey,

Just a few issues regarding further development of the project.

If the individual teams could use the coming vacation to finish off their share of the instructions (that includes testing by the way!) that would be great.

Then Annie, if you could specify in more detail the outline of the graphical user interface, and divide up the work over teams D/A and G/A. Then the individual teams should send a detailed design specification of their parts of the GUI to the 2BA7 list, so the rest of us can react to it, maybe give some 'constructive criticism', etc - so all the fine details about the GUI are known to everybody at the end of the vacation.

Don't worry, I do realize Schols are coming up (don't I know it! :) but that still gives us some time towards the ends of the holidays (after the Programming exam you'll have studied everything anyway so the practicals should be too much extra work).

Edsko

1.32 Good Luck! (sent by laura)

(Sent at Sun Mar 10 23:30:02 2002)

Heya,

Just wanted to wish yas good luck for the exams. You'll be GWATE!! :)

Regards,

Laura

1.33 RE: Some issues (sent by Annie)

(Sent at Tue Mar 12 20:33:02 2002)

Hi,

Regarding the GUI, I've spent a few days getting used to QT - there are some cool things you can do with it apparently. Unfortunately I can't get the hang of it enough to be a viable leader in this part of the project. Would anyone else be interested in doing/helping with this part?

Annie

of

1.34 RE: Proposed Work Plan (Vague) (sent by Annie)

(Sent at Wed Mar 13 18:18:01 2002)

Hello,

You can have a look at the GUI picture as in the spec report sent to Manzke at Edsko's address (I presume its still there)...

Size of Areas

Total Screen Size : 800 x 600

Program : 600 x 350

Stack : 200 x 400

Help Stack : 400 x 75

Memory : 400 x 100

Decoded Instruction : 400 x 75

Registers : 400 x 250

Possible Team Allocation

Gerard and Alan : Program, Decoded Instruction

Diarmuid and Annie: Registers, Stack, Memory

Laura and Edsko : Visualization of help stack

Other tasks : MenuBar, Menu

If anybody has any comments, suggestions or complaints with regards to the above vague outline, then feel free to post them to the list. As for the actual skeleton code, I'm going to spend some more time with QT, but Diarmuid suggested that maybe we meet up next week after the software exam, and have a look at what we can do or not do.

Annie

1.35 RE: Some issues (sent by Edsko)

(Sent at Wed Mar 13 21:22:01 2002)

hang

Hey Annie,

I would be happy to assist you on the technical details; if that's any good, let me know :-P

Unfortunately I don't have time until after the last Schol exam, programming (next Tuesday), but after then I do have time (I'm not to worried about the practicals).

Just a quick comment on your GUI suggestions (didn't scrutinize them yet :-); just that it's probably better to specify window sizes as percentages of the current monitor resolution rather than absolute pixel values. I'll comment on the rest when I've got time again :)

Keep up the good work

Edsko

1.36 RE: Some issues (sent by Annie)

(Sent at Thu Mar 14 13:48:02 2002)

Yea that would be great Edsko.... I'll talk to you more about it next Tuesday.

Annie

1.37 2ba7 project (sent by laura)

(Sent at Tue Mar 19 19:50:02 2002)

Hey,

Was just wonderin if someone could mail me how to access our source code on matrix. I tried to at home, but couldn't. Will have it updated (i.e the carry and overflow flags) by tomorrow.

See yas soon.

Regards,

Laura :)

1.38 RE: 2ba7 project (sent by Edsko)

(Sent at Tue Mar 19 19:54:01 2002)

carry

Hey Laura,

The easiest way is to access it via Chora:

<http://matrix.netsoc.tcd.ie/horde/chora>

Let me know if you have any problems,

Edsko

1.39 (no subject) (sent by laura)

(Sent at Wed Mar 20 20:29:02 2002)

Hey,

I updated the sub instruction by copying the file from matrix and working on it from my u-drive. Wasn't sure how to do it on CVS. I attached the updated file anyway :)

Good luck in the practicals!

(Skipping MIME encoded file . . . Ok)

1.40 2ba7 proj (sent by laura)

(Sent at Thu Mar 21 12:23:02 2002)

Hey,

Am just tryin to test the updated version of the instruction file. I just can't remember how to link the asm files with our program. i.e my asm file is compiling fine but when i run our program, it doesn't carry out the instructions in the h68 file. Can anyone help?

Thanks again,

Laura :)

1.41 RE: 2ba7 proj (sent by Edsko)

(Sent at Thu Mar 21 13:02:02 2002)

You have to specify the command line parameters in Visual C++. For example: test.h68 0x400 where 0x400 is the initial PC. You should be able to find this settings in the project settings.

Edsko

1.42 gui files (sent by Annie)

(Sent at Sun Mar 24 12:15:01 2002)

Hello,

Was wondering how we're supposed to access the new GUI files off CVS?

Annie

1.43 RE: gui files (sent by Edsko)

(Sent at Sun Mar 24 17:52:01 2002)

Does cvs update not give you them? Alternatively you could access them at <http://matrix.netsoc.tcd.ie/horde/chora>

Edsko

1.44 Re: gui files (sent by Gerard)

(Sent at Sun Mar 24 19:26:01 2002)

ok, you need to use "cvs update -PAd"

It's because the GUI files are in a new directory.

P gets rid of any directories that were removed in the cvs repository.

A ensures that you get the main development version of the files.

d gets any new directories in the repository, like the GUI folder! :-)

Gerard.

1.45 Friday 4th April status. (sent by Diarmuid)

(Sent at Fri Apr 5 12:39:01 2002)

hello,

Due to the fact that four of our team members participated in the foundation scholarship examinations and the fact that it was a BREAK, we were unable to make rapid progress in the project during the holidays. It should be noted that the other two members of the group did continued work on Phase 1., Also the group came together during the exams to design concept for the GUI to facilitate the progress of Phase 2.

During the last week, the team consisting of Laura and Edsko started the implementation of the help stack and integrated the Phase 1 code into the design for Phase 2. The team consisting of Annie and myself have been working on improving the design of the GUI.

The graphical user interface currently consists of a number of tables which correspond to the distinct elements of the interface as defined in the specification report. Also a number of buttons allow the user to control the state of the program. Event Handlers have been defined for every part of the system but still need to be completed. Also a number picture were created to be placed on top of the buttons and the canvas to make the interface look less technical and more accessible to the user.

The implementation of the help stack, currently detects these possible errors in the user code.

* Address error.

Bus error

Unequal number of pushes/pops in a subroutine.

An instruction such as `move $2000,a0` will give a warning (probably the user wanted to execute `move #2000,a0` instead)

An instruction such as `move a0,...` will give a warning (probably the user wanted to execute `move (a0),...` instead)

In the case of a possible error the help stack will suggestion an explanation why the error could have happened.

We are hoping to have a basic prototype of the GUI by the time of the presentation next Thursday. This prototype will feature loading files, executing the code, displaying the contents of the registers, and visualization of memory and the helpstack.

—

Diarmuid Power

Customer Relations

1.46 Compiler errors (relating include and lib files) (sent by Edsko)

(Sent at Fri Apr 5 13:33:03 2002)

Hey,

If you install Qt to a machine, the installer sets a number environment variables, amongst others QTDIR. If you then create or open a Qt project, some of the project settings make use of this variable. All fine.

However, if you make use of an existing installation of Qt on the machine, or if you accidentally told the installer not to set the environment variables, VC++ is going to give an error when you try to compile the program (something to do with include files and lib files).

To solve this problem, go to "Project", "Settings", "C/C++", "Project Options". This is a little textbox with a load of compiler command line options. Find a text saying "\$QTDIR" and replace it with "D:\Qt" (the quotation marks are only for clearness and should not be included. This assumes that you installed Qt to "D:\Qt". Now click on the tab "Link" (next to "C/C++") and do the same in the textbox "Project Options". (Two occurrences, only one for the C/C++ settings).

Finally you need to copy the file "qt-nt230nc.dll" from D:\Qt\Bin to the directory holding your project files (eg. M:\doc\M68sim).

If you then rebuild the project it should compile and run fine. You only have to do this once, it will be saved in your project settings.

Hope this helps (know it will :)

Edsko

1.47 The brand spanking new GUI (sent by Diarmuid)

(Sent at Fri Apr 5 18:30:02 2002)

hey all,

Edsko has done some fantastic work firming up the GUI design and it is looking pretty good at the moment. What he did was that he ignored the GUI designer and sorted everything out in code himself. Meaning that the GUI designer is obsolete. If you start using it for the project it might cause so problems so we encourage you to do it by hand (you will have greater control). All files have been updated properly in CVS as of this afternoon but what we must avoid is old files being used. So what you need to do when you are going to work on the project next time is:

1. Go to your cvs directory where the M68ksim files are stored.
2. Delete the directory either in windows or by typing "rm -rf M68ksim"
3. Then typing "cvs co M68ksim" which will get you a fresh copy of the

current files and you are ready to go!

if you have any problems feel free to talk to edsko or myself.

Diarmuid

Customer relations..... person

1.48 Instruction-to-string translation (sent by Edsko)

(Sent at Sat Apr 6 21:33:02 2002)

Hey all,

I set up the framework for the instruction-to-string translation (see CVS for details). I added a new method to the CPU class, Translate():

```
unsigned long Translate( unsigned long address, char *buf )
```

HOW DOES IT WORK

This function works just like Step(). However, it doesn't use the PC, but 'address'. buf should point to a 32-byte buffer. Translate() returns the address of the next address in memory (note that this is not necessarily the next address in the execution flow of the program).

For this to work, I duplicated all the methods for the individual instructions in three new files, TInstrA-C.cpp, TInstrD-N.cpp and TInstrO-Z.cpp (the names should be self-explanatory by now ;-). I changed the headers of the function to match the header of Translate(), and prefixed the name of the functions with "tS_". Eg for sub:

```
unsigned long tS_sub( unsigned long address, char *buf )
```

NOTES ON THE ts_xxx FUNCTIONS

These functions should not manipulate the PC anymore but return the address of the next instruction instead. So a typical instruction might be:

```
pc += 2;
```

This should be replaced by

```
return address + 2;
```

Note that a function such as JMP which would normally change the PC to the address in the JMP instruction should not return that address but the address of the next instruction in memory.

If an instruction is not recognized, Translate() will return -1. Therefore most calls to illegal() to indicate an illegal instruction should now return

-1 (I replaced these in the files already).

If a method declared a local variable 'address', there was a conflict with the formal parameter 'address' and the local variable. I changed the formal parameter to 'addr' in those cases.

WHAT NEEDS TO BE DONE?

First of all, make sure that if the method uses the PC to index the memory, it should use 'address' instead.

Now, typically one of these methods look like

```
if( ... )
{
xxx
}
else if(...)
{
xxx
}
```

etc. Now the parts between the braces (xxx) which is the actual execution of the instruction should be replaced by a single line outputting the correct string. Example for a move instruction for a memory address to a register:

```
sprintf( buf, "move $
```

The function 'sprintf' allows to print to a string as one would print to the terminal. The second parameter is the string to be printed. Elements starting with position in the string, and position in the string. Many more commands are available, feel free to ask me if you need one specific one. We still need to know which (long) integers to print; these are the third and fourth parameter (in this example). In general you will have 2 + x parameters to sprintf() for x Note that 'addr' and 'reg_num' in the example are (supposed to be) local

variables containing the address of the move instruction and the number of the destination register (the move() instruction will have extracted this information from the instruction encoding at some stage). The result of this command will be (for addr = 2000, reg_num = 5)

```
move $2000,d5
```

as expected. Oh, one more command will be useful: operands); these two commands will print integers in hex.

It is the responsibility of the individual pairs to complete their T.. file.

Feel free to email me if you are unsure about anything.

Edsko - your lead programmer ;-))

1.49 NOTE: Code update (sent by Edsko)

(Sent at Sun Apr 7 10:22:02 2002)

Hey,

I updated the code for nearly all CPP files on CVS to get rid of a (quite large ;) number of platform incompatibilities. Note also that GCC gave a number of possibly valid warnings to be resolved:

MInstrA-C.cpp: In method 'void M68300::chk()':

MInstrA-C.cpp:2024: warning: comparison is always false due to limited range of data type

MInstrA-C.cpp: In method 'void M68300::cmpm()':

MInstrA-C.cpp:2280: warning: comparison is always false due to limited range of data type

MInstrA-C.cpp:2287: warning: comparison is always false due to limited range of data type

MInstrD-N.cpp: In method 'void M68300::divsWord()':

MInstrD-N.cpp:33: warning: assignment of negative value '-18' to 'short unsigned int'

Note BTW that although it is not absolutely necessary to have the full instruction set implemented, it **is** absolutely necessary to have the instruction that have been implemented are working properly.

For instruction that have not been implemented, please just update the PC with 2, and in the Translate methods, return address + 2 and write something like: "sbcd: Unimplemented" to the buffer.

Oh, one more thing: return -1 as an address is non-sense of course (address are unsigned), so I changed that to return ILL_INSTR instead (has already been changed in the code).

Edsko

1.50 Saved you'se some work again ;) (sent by Edsko)

(Sent at Sun Apr 7 11:02:01 2002)

Hey,

I changed all references to 'pc' to 'address' in the T-files. Also, I changed all returns from return 0; (which I had put it to get it to compile) to return address. So instead of having to change

```
pc += 2
```

to

```
return address + 2
```

it is now simply changed to

```
address += 2
```

```
return address;
```

(which is perfectly valid as address is pass-by-value).

In most cases this will be sufficient (for the return part that is)

Edsko

PS. Very excited about Qt :)

1.51 How to test the T.. files (sent by Edsko)

(Sent at Sun Apr 7 12:41:01 2002)

Hey,

I added a new option to the phase 1 main(). You should now execute the program with three command line parameters:

```
68sim file.h68 init_pc [exec—decode]
```

The last parameter (either 'exec' or 'decode') tells the program to run Step() (exec) through the program or Translate() (decode). So if you have a testfile test.h68, with initial PC 0x400, and you want to test your decoding instructions, you would invoke

```
68sim test.h68 0x400 decode
```

Or if you wanted to test the execution instead

```
68sim test.h68 0x400 exec
```

I made an example test file, 'testsub.asm' (can be found in the testfiles/ directory on CVS). If you run the compiled version of this program through 68sim, you should get the same instructions as in the test file (I finished the decoding on sub). This should give you an idea on how to do the decoding (look at the tS_sub in TInstrO-Z.cpp).

One more note: we have to decide whether to output numbers in hex or in decimal. I'd suggest that for now we keep the convention of outputting all numbers of decimal numbers, except for addresses (in hex).

It doesn't take much time to write these methods when the decoding is already done. Simply a matter of removing the instructions that execute the

instruction and replace them by a single `printf()` call.

Edsko

1.52 Number format in T files (sent by Edsko)

(Sent at Wed Apr 10 14:47:02 2002)

Hey,

Just as a reminder, please stick to the convention of displaying all numbers in decimal, except for addresses which should be printed in hex. (I would also be happy if all values are displayed in hex, but definitely don't print addresses in decimal). Also, indicate hex numbers with a leading \$, ie. to print `d[0]`:

```
printf( buf, "$  
Edsko
```

1.53 Compiling for console mode (phase1) (sent by Edsko)

(Sent at Thu Apr 11 15:33:02 2002)

Hey,

There was a small problem with the using the same code for phase 1 and phase 2.

The `Translate()` method on the CPU didn't execute the `Halt()` for \$F001 (just outputted the decoding); this is the correct behaviour of course.

However, in console mode (and in particular, in console decode mode), the `Halt()` should be executed, since otherwise the program will never ever stop decoding memory...

I fixed this in the following way. If you're compiling to the the GUI, everything is fine and no worries. However, if you're compiling for phase 1, you have to add the following compiler option (in "C++" option in VC++, or just as an extra command line parameter in *NIX):

```
-DPHASE1
```

this will make sure the console mode of the program works fine as well.

Note that to compile the program in console mode in Solaris this needs a slightly more complicated command:

```
g++ -o 68sim -fno-operator-names /cvs/M68ksim/*.cpp /usr/lib/libm.so -DPHASE1
```

Hope this is clear,

Edsko

1.54 moc files and cvs (sent by Edsko)

(Sent at Thu Apr 11 18:26:02 2002)

Hey,

I removed the MOC files from the CVS repository as they will be autogenerated by the Qt precompiler. Therefore if CVS is complaining something like

? GUI/moc_blah

Just ignore that.

(It doesn't make too much sense to store something in a CVS repository that is autogenerated)

Edsko

1.55 Final demo ideas (sent by Edsko)

(Sent at Thu Apr 11 18:27:02 2002)

Hey,

Please reply to this email with ideas on how the final demo should be generated.

I suggest Diarmuid and I will do the demo again with the same division of tasks as we have done so far.

Then Annie and Laura could concentrate on generating the user manual for the program, and Gerard and Alan could concentrate on generating the other (technical) of the project.

Edsko

1.56 DEADLINE! DEADLINE! DEADLINE! (sent by Edsko)

(Sent at Thu Apr 11 18:40:01 2002)

Hey,

Just to remind all of you, Friday the 19th of April is the DEADLINE FOR PHASE THREE. That means that the instruction decoding, instruction execution, and graphical user interface need to be COMPLETED BY THEN.

In the week after that we can add some colour to the GUI, pictures, maybe toolbar, stuff like that. I will try to implement backward stepping over the next weekend, not sure if I'll get that completed but I'll try.

NOTE THAT "COMPLETED" IMPLIES *TESTED*.

Edsko

PS. "Debugging is twice as hard as writing the code in the first place. Therefore, if you write the code as cleverly as possible, you are, by definition, not smart enough to debug it."

1.57 Discussion of the product. (sent by Diarmuid)

(Sent at Thu Apr 11 18:53:01 2002)

Hello,

The back-end of the simulator has advanced to a stage where it can be successfully demonstrated. Also the graphical front-end is now nearly functionally complete. It might be useful to organise a short meeting so we could show you the program, and see if you're happy with it (I'm sure you are but still :-), especially since we are running towards the product release date (early May) very rapidly.

Let me know if you would be interested in this,

Diarmuid Power
Customer Relations

1.58 Status report (sent by Edsko)

(Sent at Fri Apr 12 11:21:01 2002)

Hi,

The graphical user interface to the program is now nearly complete, with the exception of the visualization of the system stack (which is still a reasonable challenge) and the opcode bitmap (minor task). We still have to improve the visual appeal of the GUI (we have concentrated only on the functionality so far).

We realized that we had forgotten to translate the instruction from binary to human readable strings (even though we have the execution engine done), so we started work on that and have progressed a good bit.

The original deadline for phase 2 (development of the GUI) was set to today, and the deadline for phase 3 (context sensitive help system) was set to next Friday; however, we worked on both parts of the program during the last few weeks and we have decided to merge the deadlines to one deadline for both phase 2 and phase 3; this deadline is still set to next Friday (19th).

We started writing test files for the individual instructions so we can effectively test the program.

Coming up is the development of a 'backward step' (or 'undo' function), improvement of the help stack (detect more errors), completing the decoding and execution engine and most of all, testing the program.

Edsko de Vries
Lead Programmer

1.59 Some stats on the code. (sent by Diarmuid)

(Sent at Fri Apr 12 19:31:02 2002)

hey all,

Here are some quick stats on the code that has been written so far:

This is for the .h and .cpp code:

17953 lines of code

197 lines of comments

3159 lines are blank

21309 lines in total

and the test files that we have written:

346 lines of code

0 lines of comments

96 lines are blank

442 lines in total

Keep up the good work team!

Diarmuid Power

1.60 QT stuff (sent by Alan)

(Sent at Sat Apr 13 18:55:01 2002)

Hey folks,

Hope no one has implemented this already but I was getting a little bored with instruction decoding and decided to have some fun with Qt :)

I've implemented What's This functionality in the program, so now, when you click on the question mark button at the top right and click on a control, it gives up a 'what's this' help message beside the control you clicked that's specific to the particular control. (For example, clicking on the memory viewer, pops up a message saying 'Memory Viewer Shows the contents of memory', or something similar...).

I've also implemented the opcode bitmap loading. You just need to have a bitmap saved in a format QT supports (.gif and .jpg work fine) in the same folder as the program called <instr_name>.gif or <instr_name>.jpg where <instr_name> is the instruction method's name. They're called using a modified version of LINK and LINK_EXT macros, Edsko, is this OK or should they be called from the end of each routine?

Unfortunately, because I'm using Windows 2000 Offline Files to synch between the laptop and matrix, I probably won't be able to upload the files till monday as matrix doesn't allow Samba connections from outside college for some reason :(I might risk SCPing them to matrix and updating CVS and cross my fingers that windows won't mangle the files when I sync. again...

Anyway, hope it was OK to go ahead and do the above :)

Alan

=====
adonnell@tcd.ie

1.61 RE: QT stuff (sent by Edsko)

(Sent at Sun Apr 14 10:35:01 2002)

with
memory',

So far so good :-) A useful thing to do; Thanks :)

modified
some

But... Diarmuid and I also implemented this ;-). Well, that is to say, we loaded one (as a prove of concept). We should compare your solution and our solution and determine which one is the best... However, we hadn't implemented the actual linking between instructions and the bitmaps so that work is definitely useful :) One comment: JPG is a licensed format and Qt doesn't support that on all platforms. Therefore we settled for PNG instead. This is an open standard and implements good compression (better even than JPG).

We'll talk about this on Monday, so don't upload yet :)

Edsko

1.62 RE: QT stuff (sent by Edsko)

(Sent at Sun Apr 14 10:37:02 2002)

(Forgot to say this: the version on CVS actually supports the opcode bitmap loading. You could have a look at it and see what you think. Unfortunately I can't look at your solution :)

1.63 Code changes (sent by Edsko)

(Sent at Sun Apr 14 12:54:01 2002)

Hey,

I made quite a large number of changes in the code to solve a number of Qt connection issues. I also changed MemDevice and M68300. One of the changes is that the simulator now tries to make an 'educated guess' about the initial pc of the system. So you can now start your testfiles at \$400 and it will pick that up automatically. Also, I added a Reset() method to the CPU class, and the reset button is connected to that method (this restores the registers to zero, sets the pc to the initial pc, a7 to the initial SSP, and clears the halt status of the CPU, but doesn't clear memory).

So it is now more important than ever that you run cvs update first.

Next, for those who want to try to compile the program in Linux, I wrote a small shell script to do this; see attachment.

Will start working on Undo now :)

Edsko

(Skipping MIME encoded file . . . Ok)

1.64 Small issue: 68sim.cpp (sent by Edsko)

(Sent at Sun Apr 14 13:09:01 2002)

Hey,

If you are compiling the code for the GUI, you probably know that you shouldn't include 68sim.cpp (as this has the main() for the console mode program). Now this is no big problem in VC++ (simply remove it from the project). However, when compiling on the command line this is a bigger pain in the neck.

Therefore I changed it; you can simply leave it and compile it; it will have no effect unless you use the -DPHASE1 option that I described earlier.

Edsko

1.65 suggestion for help stack. (sent by Diarmuid)

(Sent at Sun Apr 14 13:23:02 2002)

hey all,

This is particularly relevant to Laura and Edsko with their work on the help stack but I was thinking about the System stack yesterday and I was thinking that if the user tries to defy the stack growth convention then the help stack should scold them.

Example

A sequence of instructions.

```
move.w #255,-(a7) Fine
move.l d0,-(a7) Fine
move.l d7,(a7)+ Help stack warning.
```

```
move.w (a7)+,d0 Fine
move.l -(a7),$400 Help Stack warning.
```

if the convention is adhered to then this will catch any mistakes that they make. And it should be minimal in code too.

Diarmuid

1.66 Code changes (2) (sent by Edsko)

(Sent at Sun Apr 14 17:37:02 2002)

Hi,

Undo now works, as does Run/Stop. Also the instruction currently pointed to by the PC is highlighted.

That summarized a lot of work in two sentences :-)

Edsko

1.67 Re: Final demo ideas (sent by Diarmuid)

(Sent at Mon Apr 15 18:22:01 2002)

On Thu, Apr 11, 2002 at 06:26:36PM +0100, Edsko de Vries wrote:

Will we have to have a section on how the team worked(minor), what the design was about. How the product is a teaching tool. We will have the actual demonstration in there somewhere. The focus I believe will have to be as a teaching tool. We will have to come up with use cases for the demonstration. Organise that it will be demoed on someones laptop and decide that in advance and maybe test the projector that it is going to be used on.

Diarmuid shudders at the thought of speaking.

1.68 RE: Final demo ideas (sent by Edsko)

(Sent at Tue Apr 16 13:45:02 2002)

Hi,

How about:

We concentrate on the difference between our program, a teaching tool, and an ordinary debugger. We show off the help stack (with mainly `_possible_` errors, such as: you probably didn't want to do that), stack viewer (with focus on the origin) and undo functionality; e.g. we have a BEQ instruction, we take the instruction and branch, then we undo the branch, change the status of the flags, execute the instruction - and we don't take the branch! Hence showing off the effect of the flags on the branch instructions.

Edsko

1.69 Documentation? (sent by Edsko)

(Sent at Tue Apr 16 19:21:02 2002)

Hi,

We were wondering what you would like to receive as documentation of the project. What should be included in the documentation? What form should the documentation be in? (Printed or electronic)? Do you need a code dump? (A printed code dump would take approximately 500 pages...) A user manual maybe?

We would appreciate if you could give us some details on this so we could start working on it...

Thank you

Edsko de Vries
Lead Programmer Group B

1.70 System stack (sent by Edsko)

(Sent at Wed Apr 17 10:38:02 2002)

Hi, and mainly: Hi Diarmuid/Annie

First of all, I won't be in today, don't feel very well.

Secondly, I found some difficult problems with the system stack display. Have a look at the first three instructions of hstack.asm (execute them) and you'll see what I mean. Now, I solved it for this particular situation (look closely at the stack table while you execute the instructions), but it doesn't work for the other way around (word first then long, look at the file to see what I mean).

Now, have a look at my solution in MInstrD-N/move. It works, but... Big problem: Undo.

I haven't really figured out a solution, so here is something to think about for you :) One thing you could do is save the information about the contents of the stack per byte in some applied data structure; that would solve the problem. You will have to change the structure quite a bit for this though. (Again, think about the problem first, then you'll know why this is a solution). Not sure if this is the best solution though.

Good luck :)

I'll see you tomorrow

Edsko

1.71 RE: System stack (sent by Edsko)

(Sent at Wed Apr 17 17:09:02 2002)

No, Gerard's fault... He started the cold :-)

which

Yeah, looks good. If you feel like doing SBCD, by all means, go ahead :-)

Also saw your changes in the GUI (re focus). Good work :)

Edsko

1.72 Re: Documentation? (sent by Michael)

(Sent at Thu Apr 18 08:15:02 2002)

Dear Edsko,

I cannot answer this question. I presume Declan would like all groups to document their work in similar format.

Regards,
Michael

Edsko de Vries wrote:

1.73 Re: Documentation? (sent by <osulldps@cs.tcd.ie>)

(Sent at Thu Apr 18 11:13:02 2002)

Hi Michael

I replied to Edsko yesterday but forgot to cc it to all.

I will be making the general guidelines available to all.

Thanks
Declan

1.74 Meeting tomorrow! (sent by Diarmuid)

(Sent at Thu Apr 18 17:58:01 2002)

hey,

meeting tomorrow will be pushed back to 12. Is that ok with all?

Diarmuid

P.S. Diarmuid will be joining you remotely as he has places to be.

1.75 Friday 19th April update (sent by Diarmuid)

(Sent at Fri Apr 19 12:30:02 2002)

Hi,

Today is the deadline for Phase 3 which consisted of the completion of both the backend and the graphical front end. All pairs have met this deadline or are very close to meeting this deadline. The next week will be spent finishing off the last bits of coding, and we will be testing the program thoroughly by writing programs and testing a large variety of instructions and the most common instructions.

Next Friday will be the closing date for code changes. From then on, we will be preparing for the demo, cleaning up the code and starting to write the documentation.

Diarmuid Power
Customer Relations

1.76 gcc warnings (sent by Edsko)

(Sent at Fri Apr 19 15:44:03 2002)

For those of us who only use VC++:

```
/home/edsko/cvs/M68ksim/MInstrA-C.cpp: In member function 'void
M68300::add()':
/home/edsko/cvs/M68ksim/MInstrA-C.cpp:661: warning: unreachable code at
beginning of switch statement
/home/edsko/cvs/M68ksim/MInstrA-C.cpp: In member function 'void
M68300::bra()':
/home/edsko/cvs/M68ksim/MInstrA-C.cpp:2912: warning: comparison is always
false
due to limited range of data type
/home/edsko/cvs/M68ksim/MInstrA-C.cpp: In member function 'void
M68300::bsr()':
/home/edsko/cvs/M68ksim/MInstrA-C.cpp:2970: warning: comparison is always
false
due to limited range of data type
/home/edsko/cvs/M68ksim/MInstrA-C.cpp: In member function 'void
M68300::chk()':
/home/edsko/cvs/M68ksim/MInstrA-C.cpp:3037: warning: comparison is always
false
due to limited range of data type
/home/edsko/cvs/M68ksim/MInstrA-C.cpp: In member function 'void
M68300::cmpm()':
/home/edsko/cvs/M68ksim/MInstrA-C.cpp:4129: warning: comparison is always
false
due to limited range of data type
/home/edsko/cvs/M68ksim/MInstrA-C.cpp:4140: warning: comparison is always
false
due to limited range of data type
/home/edsko/cvs/M68ksim/TInstrA-C.cpp: In member function 'long unsigned int
M68300::tS_bxx(long unsigned int, char*)':
/home/edsko/cvs/M68ksim/TInstrA-C.cpp:2130: warning: comparison is always
false
due to limited range of data type
/home/edsko/cvs/M68ksim/TInstrA-C.cpp: In member function 'long unsigned int
M68300::tS_bra(long unsigned int, char*)':
/home/edsko/cvs/M68ksim/TInstrA-C.cpp:2387: warning: comparison is always
false
due to limited range of data type
/home/edsko/cvs/M68ksim/TInstrA-C.cpp: In member function 'long unsigned int
M68300::tS_bsr(long unsigned int, char*)':
```

/home/edsko/cvs/M68ksim/TInstrA-C.cpp:2453: warning: comparison is always false
due to limited range of data type
/home/edsko/cvs/M68ksim/TInstrA-C.cpp:3155:1: warning: no newline at end of file

1.77 Bugs and bugfixes, also explanation of I/O (sent by Edsko)

(Sent at Sat Apr 20 11:31:01 2002)

Hi,

I starting testing the program this morning by writing a program that input a string from the user, then reversed it, and output it to the user again.

Okay, so far I/O. On to the main point of this email; I immediately found a number of bugs:

MOVEA.L # $\$800$,a1

Even though the translation seem to work fine and gives a 6-byte opcode, the execution doesn't; the PC goes from 400 to 404 instead of 406.

a1 is not affected

A quick look through the code indicated that MOVEA doesn't handle any immediate values longer than 1 byte (?)

LEA $\$800$,a1

Again, though the translation works fine, the execution doesn't. PC gets updated correctly, but the value which gets loaded into a0 (which should be $\$800$ of course) is some random very high value, and not $\$800$. I'm not sure if this is helpful, but note that the difference between movea $\$800$,a0 and lea $\$800$,a0 is that move will use $\$800$ as a pointer, whereas lea will push the actual effective address (ie. $\$800$).

BSR -FIXED-

The translation for BSR adjusted the SSP (a7) causing a lot of problems :)
BSR/RTS now works again

SUBQ -FIXED-

The manual specified that SUBQ need not handle ax operands. However, the assembler seems to think otherwise so I added this new functionality to SUBQ (both M and T).

Then I couldn't really continue testing anymore as I found that CMP (T) hadn't been implemented. As we need CMP in practically every application, this is a high-priority instruction :) Please submit this asap. Thanks :)

--

BTW, maybe this is a good time to explain the I/O features I designed:

dc.w $\$F002$

This instruction reads a string from memory, starting at position a0, and opens a messagebox with this string. Note that you can use several HTML tags to format your strings (!) :)

dc.w \$F003

This instruction opens a input dialog box with a single textfield. When the user enters a string into the textfield and clicks Ok, the string will be stored in memory, again starting at position a0. The input dialog box will also contain a message (saying for example: "Please enter a string" or something similar). This string is also contained in memory, starting at position a1. Again, HTML can be used in this message (eg
 if you want to continue on the next line).

Edsko

1.78 Bugs and bufixes, and cool tutorial program (sent by Edsko)

(Sent at Sun Apr 21 11:06:01 2002)

Hi all,

After a number of more bugs and bugfixes, I got the first "real" program to work in our simulator! See testfiles/reverse.asm. This program reverses the string that the user enters and outputs it again. Load it into the simulator, and just press RUN, and watch the show :)

Okay, for the bugs (one not fixed):

Bxx (fixed)

If the branch was not taken, PC was not updated

SUBQ (fixed)

subq #imm,an was never executed and was written slightly careless ;)

Step() vs. I/O (fixed)

If you pressed run, and the program had some I/O call, say, read string, the program tried to continue

while the user had to enter a string. The step button now stops the timer,

runs Step() on the CPU and

starts the timer again (if applicable)

ADDQ -NOT- FIXED

addq #1,an doesn't seem to update the PC

CMP (fixed)

cmp an,an followed by a BLE doesn't work. Found the problem in Bxx which used bitwise and/or (—/&) instead of

logical and/or (—/&&). The following regexp fixed this in one go:

s/\([—&]\)/\1\1/g :-)

Edsko

1.79 Bugs and bugfixes, suggestion (sent by Edsko)

(Sent at Sun Apr 21 13:37:02 2002)

Hey,

First of all a suggestion: I suggest that we put everything in hex. Every output, every input, consistently in hex. I find it a bit annoying the way it's currently working. If you guys agree with this I'll change it myself (regexps :)

Okay, as for the bugs:

Bxx (fixed)

Small mistake in BLE (a && should be a —, was correct in the comments but not in the code itself :)

Open (fixed)

Open didn't reset the CPU

move, T -NOT FIXED-

Translation of move has a few bugs. Some \$'s are missing, and where there are \$'s, the number is in decimal, not hex

(execution is fine)

addq -NOT FIXED- (not sure if I already specified this bug)

Doesn't update PC in most cases.

addi -NOT FIXED-

Doesn't update the PC

add -NOT FIXED-

The assembler seems to think that AND also supports immediate operands and `#$7FFFFFFF,d0` gets compiled as an AND instruction, whereas `andi #$7FFFFFFF,d0` gets compiled as an ANDI instruction (according to the translations anyway)

Edsko

1.80 Help stack problems (sent by Edsko)

(Sent at Mon Apr 22 13:02:01 2002)

Diarmuid (cc: other group members :-)

The main problem with the help stack currently is best explained with an example:

```
move.l d0,-(a7)
move.w (a7)+,d0
move.w (a7)+,d1
```

As you will realize this won't work with our original solution. Now, I changed this, and my changes solve this particular problem. However, it won't work with undo, and it won't work either in the following case:

```
move.w d0,-(a7)
move.l (a7)+,d1
```

Have a look at the code to see what I have done.

What we probably should do is if we push something like

```
move.l d0,-(a7)
```

We should push 4 bytes to the system stack as opposed to 1 longword. That way we can pop off 2 bytes, 1 byte, etc. Also, popping off from the system stack unto the undostack (mstate, remember) would work properly then.

If you have any problems, let me know :)

Edsko

1.81 ADD and AND Issues (sent by Alan)

(Sent at Mon Apr 22 15:49:01 2002)

Edsko and Annie very kindly (!) pointed out some problems with the ADD(Q/I/X) and AND(I) routines.

I spend the weekend testing these instructions and making appropriate modifications and they should all work properly now :)

ASL_R to follow shortly...

Alan

```
=====  
adonnell@tcd.ie
```

1.82 Mailing List Archive PDF (sent by devriese@tcd.ie)

(Sent at Mon Apr 22 17:04:01 2002)

Hey all,

I wrote a C program that takes the raw source of the 2ba7-group-b mailing list archive and translates it into Latex.

I ran it on the archive as it is now. The result can be viewed from <http://matrix.netsoc.tcd.ie/~edsko/68sim/mbox.pdf>

Let me know if you have any suggestions for improvement :)

Edsko

1.83 Makefile and GCC warnings (sent by Edsko)

(Sent at Tue Apr 23 13:30:02 2002)

Hi all,

For those of you who are compiling the program on *NIX, I wrote a Makefile for the project which finally enables incremental compilation (ie. you don't have to compile the whole project for every small change, but it will only recompile the parts that need recompilation).

Just create an empty directory on your Matrix drive, copy the Makefile into that directory, and type "make". That should compile the whole project. All files that are created in the process (.o, moc files, the final executable etc.) will be stored in the current folder, to keep the CVS folder clean.

The next time you run make, it will see if it needs to compile anything, and if so, it will recompile that part of the project which is much much faster. Nice, er? ;-)

Also, I changed the options to the GCC compiler, to turn on full warnings - this resulted in quite a substantial list of warnings. I made this list available at <http://matrix.netsoc.tcd.ie/edsko/68sim/warnings.txt>.

Edsko

(Skipping MIME encoded file . . . Ok)

1.84 Makefile for Solaris (sent by Edsko)

(Sent at Tue Apr 23 16:12:01 2002)

Hey all,

I have spent the last five hours fixing bugs in the program... The most difficult one to fix had to do with the fact that Solaris runs on a little endian machine, and I hadn't taken that into account in the definition of the registers. But I finally fixed it (very happy :).

I attached a new Makefile. This Makefile is *_only_* necessary for Solaris (ie. Matrix). For big-endian *NIX (such as Linux on PCs) use the original Makefile. (The only difference is a `-DLITTLE_ENDIAN` option which is interpreted in M68300.h)

Edsko

(Skipping MIME encoded file . . . Ok)

1.85 GUI Update (sent by Gerard)

(Sent at Tue Apr 23 23:21:01 2002)

Hey all,

ok I've changed the GUI so that it can be resized, minimised, maximised etc. Unfortunately, somehow lost the Context Help button along the way! Gerard shrugs*

Don't know why, it should be there, I strongly suspect Microsoft!

Here's a dump from the CVS log: (slightly edited with hindsight)
Ok, changed the GUI so it's not confined to a specific size.
Not perfect yet, some slight bugs when you make the window quite small, not sure if it's my code or just Qt in general.
Also not happy with the bottom right corner. Is anything else being added here?

Still has minimum sizes set so that it can't be made to be too small and hence unusable.

Also added minimise and maximise buttons to Windows version however, the Context Help button has been lost. Don't know why it won't show up, it's set to be on.

It now starts the program maximised rather than just with the normal exec() command.

Moved IR, PC and CCR displays to the right, giving more room to System Stack. Made LEDs slightly bigger as well.

Gerard

1.86 RE: bugfixes (sent by Annie)

(Sent at Wed Apr 24 11:24:02 2002)

Hi,

Have fixed the errors below and tested them and they are working fine I think.

the

A

1.87 RE: GUI Update (sent by Alan)

(Sent at Wed Apr 24 13:44:01 2002)

Don't worry about the What's This Help. Since this was my idea, I'll either implement a popup menu for when you right-click with a what's this entry on it or have some short-cut key combination.

It'll probably be the former as QMTable seems not to respond correctly to keyPressEvent events...

Alan

=====
adonnell@tcd.ie

1.88 PDF of CVS log (sent by Edsko)

(Sent at Wed Apr 24 16:44:01 2002)

Hi all,

I wrote another C program that took the log from the CVS file and loaded it into PostgreSQL. I then wrote a PHP script which took this out of the SQL database again, but sorted by date (rather than sorted by file which is the default for the CVS log), and outputted this to a Latex file. The Latex file of course translated into a PDF.

This PDF is available from
<http://matrix.netsoc.tcd.ie/~edsko/68sim/cvslog.pdf>.

I think that this PDF along with the PDF of the Mailing List archive will give a good overview of the history of the project. Declan can't complain :)

Edsko

1.89 PDF of CVS log (sent by Edsko)

(Sent at Wed Apr 24 16:48:01 2002)

Ok, as Gerard was so keen to point out, there is a bit of redundancy in the PDF. I'll look into that :)

Edsko

1.90 PDF of CVS log (sent by Edsko)

(Sent at Wed Apr 24 16:59:02 2002)

Done (from 81 pages to 20 pages :). Updated file at same location.

<http://matrix.netsoc.tcd.ie/~edsko/68sim/cvslog.pdf>

Edsko

1.91 The final package (sent by Edsko)

(Sent at Wed Apr 24 20:33:02 2002)

Hi all,

Suggestion: The final package will consist of a single CD-ROM, which contains everything we have ever done on this project. The CD-ROM autoruns in Windows, at opens with the following webpage (now stored on my website for convenience, of course eventually we'll put a local copy onto the CD):

<http://matrix.netsoc.tcd.ie/~edsko/68sim/intro.html>

(design can still be changed ;-). Once we're happy with the design of this webpage, I'll design a matching CD-cover for the project as well, including a

CD-label (I have a CD labeler) and that's our product, nice and professional.

Ideas?

Edsko

1.92 Re: The final package (sent by Alan)

(Sent at Wed Apr 24 20:47:01 2002)

Edsko, you work FAR too hard – but you knew that already :))

I really like the mock-up you've done, however, you're going to have slight problem with the links to the executables – they'll open a 'download this program' window. Most local webpages that need to do this (on CDs etc...) use an ActiveX plugin to get around this (not compatible with anything other than IE, of course :(- you also have to pay lots for the plugin).

So if you have any more spare time, I'd imagine doing your own ActiveX plugin would keep you occupied for a while – perhaps not :) Alternatively, you could use QT's rich text viewer in a dialog, which would be cross platform compatible, and might run the setup program without asking if you want to download it from the CD. It'd also be alot easier to implement than an ActiveX control, I imagine :)

Fantastic work so far though...

Alan

=====
adonnell@tcd.ie

1.93 RE: (none) (sent by Edsko)

(Sent at Wed Apr 24 23:18:02 2002)

Er, why the apologies?

won't

No no, that's why I sent it to you guys... After all, this is a group project :) I wonder though how you "fixed" them.. Did you break into my account? :-)
If you just fixed your local copy, if you would be so kind to send the updated version to me, I'll update it to my Matrix space.

Edsko

1.94 Re: The final package (sent by Diarmuid)

(Sent at Thu Apr 25 00:33:02 2002)

I know I have mentioned it before but I would rather see the documentation on CD rather than paper. Saving the forests and all that.

Ditto to what Alan said. Great work edsko!

Diarmuid

On Wed, Apr 24, 2002 at 08:32:51PM +0100, Edsko de Vries wrote:

1.95 Hex vs. Decimal (sent by Edsko)

(Sent at Thu Apr 25 17:54:01 2002)

Hi all,

I got rid of all GCC warnings (many of which were important warnings), so you should all be thankful :)

Secondly, I wrote a VIM script that changed `_all_` references from from a \$ there, or if it was a register name (we don't want d\$4). Note that even though it sounds strange to display the register numbers in hex, as register numbers don't go higher than 9, this shouldn't be a problem.

This is the script:

```
 :::wq
```

I looks right to me (Alan? :-) The project seems to compile and work fine, but if there is any problems, let me know.

This assures that all numbers are in hex, and properly indicated as being hex.

Edsko

1.96 Deadlines, Generic stack reset() and GUI (sent by Edsko)

(Sent at Thu Apr 25 19:37:02 2002)

Hi,

First things first: The deadline tomorrow: the code as it is tomorrow will be used for the project demo. I suggest that we leave the CVS repository open to changes until Friday the week after, so that additional bugfixes / GUI elements can still be added. Then that Friday will be the final deadline, and I will create the installshield and make the UNIX tarballs from the state of the CVS repository of that Friday (let's say I will do the last CVS update Saturday morning, so we can upload changes until Friday night).

If I then could have the final documentation (user manual and technical documentation, see the slides of the last 2ba7 lecture for notes on the latter, "project report" in Declan's terminology) at the very last by Tuesday the 7th of May, so I have some time to finalize the CD's and burn them.

Then the other matters: Sorry Gerard, but the function `Reset()` which you have added to the generic stack class indeed has "some memory problems" (as you already expected on the CVS logs). Only I wouldn't have said "some", but

"huge" ;-)). I'll explain that:

Two problems: The first problem is that the stack is implemented as a dynamic linked list. The nodes which make up the stack are created (memory is allocated for them) on the fly; and they are not removed by reset(). That's the first memory problem.

The reason that I didn't implement a reset() or clear() method in the first place, is the second problem: You can implement a stack of pointers to dynamically created objects; and this is indeed done, StackItem elements for example:

```
Stack<StackItem *> systemStack;  
systemStack.push( new Stackitem() );
```

A proper clear() should deallocate these elements as well (StackItem or whatever other dynamically allocated elements). However, precisely because it is a generic stack, it cannot assume that the elements on the stack are dynamic and can therefore not delete them.

Hence a reset of the system stack or help stack needs to be done outside of the generic stack class in the code itself. I will take care of this :)

On the GUI: Good work, I like the resizing. Two small remarks: It seems that when you make the window bigger, the last column in the memory viewer (15) gets bigger than the rest; I would prefer the address column to increase with the data columns remaining the same size (as one another). Secondly, maybe we shift the space ratio between the program viewer and system stack viewer in favour of the program viewer.

Edsko

1.97 RE: Hex vs. Decimal (sent by Alan)

(Sent at Fri Apr 26 10:04:02 2002)

Seems you're right Edsko: I came up with a few reg exps. yesterday evening that allow switching constants in the program viewer (#xxx and displacements) between hex and decimal :D You can let me know what you think (it defaults to hex BTW ;)...

Anyway, I didn't really mean you to take that e-mail completely seriously - I should probably just start using VIM and leave the horrible world of Visual Studio behind me :))

Alan Donnelly
=====
adonnell@tcd.ie

Edsko de Vries <devriese@tcd.ie> wrote:

1.98 Friday 26th: Status report (sent by Diarmuid)

(Sent at Fri Apr 26 12:04:01 2002)

hello,

Today the current snapshot of the repository will be used to construct the demonstration for next Thursday. The demonstration will be prepared in advance by Edsko and myself while the rest of the team will be focusing on preparation of documentation of the project. The documentation deadline will be scheduled for next Tuesday week (7th May) to ensure that the final package is brought together in good time.

Final deadline will be next Friday night where the CVS repository will close for the end of the project. Minor bug fixes will be accepted up to that point.

Feedback on Group G

What we liked about their design:

Network streaming + XML

What they should emphasise in the final demonstration:

Ethical and moral issues involved in the application of the project.

Diarmuid Power
Customer Relations,
Group B

1.99 Stack operations (sent by Diarmuid)

(Sent at Fri Apr 26 21:03:02 2002)

hey all,

Stack operations are mostly there. We have it operating as it should mostly with both pushing and popping operations implemented. One or two quirks still remain. I have attached the test files that demonstrate the quirk and its propagation to the next push operation and if anyone can point me in the right direction before I spot it please do.

Diarmuid Power
Going to find food!

1.100 Re: Stack operations (sent by Diarmuid)

(Sent at Fri Apr 26 21:04:01 2002)

-C7zPtVaVf+AK4Oqc

I suppose attaching the test files would help. *sigh*

Diarmuid

— Forwarded message from Diarmuid Power <diarmuid@netsoc.tcd.ie> —

(Sent at Fri, 26 Apr 2002 21:02:09 +0100)

1.101 Stack operations (sent by Diarmuid)
1.102 Stack operations. (sent by Diarmuid)

(Sent at Sun Apr 28 01:46:02 2002)

hey all,

Finally got the stack operations working as they should, testing seems to work fine. I have made the necessary modifications to move and jsr so far and I hope to complete more before the end of the project. But first I intend to take a day away from computers or TV as I believe that the radiation of prolonged exposure is frying my eyeballs.

Signing off at 1:46 in the morning,
Diarmuid

1.103 RE: Proposed CD covers (sent by Annie)

(Sent at Sun Apr 28 14:05:01 2002)

Great designs Edsko.

1.104 away till tuesday (sent by Annie)

(Sent at Mon Apr 29 19:15:02 2002)

Hey y'all,

hope everything is going well. Just to apologise for my absence for the next few days - hope to back on either tuesday or wednesday.

Annie

1.105 68ksim Update (new makefile) (sent by Alan)

(Sent at Wed May 1 17:02:01 2002)

Finally got Solaris and my code to put their differences aside today, and compile in harmony :)

I've moved the slot definitions for the program options to options.cpp, so you'll need the updated makefile attached to make it compile correctly on Solaris. Main updates include:

Updated program viewer scrolling – now more robust than ever :-P

Can now switch assembler constants between hex and decimal

Help stack briefly highlights new errors

Unfortunately didn't get keyboard focussing to work correctly – some other day perhaps :)

Alan Donnelly
=====
adonnell@tcd.ie

1.106 68ksim Update (new makefile) (sent by Alan)

(Sent at Wed May 1 17:11:01 2002)

Finally got Solaris and my code to put their differences aside today, and compile in harmony :)

I've moved the slot definitions for the program options to options.cpp, so you'll need the updated makefile attached to make it compile correctly on Solaris. Main updates include:

Updated program viewer scrolling – now more robust than ever :-P

Can now switch assembler constants between hex and decimal

Help stack briefly highlights new errors

Unfortunately didn't get keyboard focussing to work correctly – some other day perhaps :)

Alan Donnelly
=====
adonnell@tcd.ie

1.107 Final deadline (sent by Edsko)

(Sent at Fri May 3 11:50:01 2002)

Hi,

Tonight is the final deadline for the project. I will take the contents of the CVS repository tonight and start creating the packages. I will also run the scripts that I wrote to convert the mailing list archives and the CVS logs tomorrow morning for the last time. This means that if you want anything to be included in either three, it has to be done before tomorrow morning.

Edsko

1.108 Status report (sent by Diarmuid)

(Sent at Fri May 3 13:04:01 2002)

hey all,

Code freezes today. Then it will begin the packaging stage.

Documentation is due on Tuesday at the latest and will be integrated into the final package before thursday when the media will be finalised and produced for submission.

Demonstration went well yesterday and all team members(except for edsko and gerard who slunk off early, we will get you later! :) went and celebrated the great achievement that the group has made.

Everything is on schedule and looking good for the final product.

Diarmuid Power
Customer Relations Representative,
Group B

1.109 CVS repository closed (sent by Edsko)

(Sent at Sat May 4 10:48:01 2002)

Hey,

The code is now officialy frozen. Uploads to the CVS repository will not be used in the final product.

Just as a reminder, coming Tuesday is the deadline for the documentation.

Edsko

1.110 Final code changes/cleanups (sent by Edsko)

(Sent at Sun May 5 10:35:01 2002)

Hey,

I have cleaned up the code and made a few additional changes (listed below). This code can now be downloaded from the CVS repository. Note that though the CVS repository will not refuse code changes, new code will not be used for the final package.

CHANGES

Changed LITTLE_ENDIAN to M68K_LITTLE_ENDIAN
Introduced a local variable unsigned short int ir; in Translate() to avoid Translate() effecting the state of the CPU
Made sure BSR and RTS work with the new system stack
Introduced POP in movea
Re-introduced the pushStack
Added the name of the opened file to the title
Removed redundant printf statements
Changed copyright notices
Changed all references from 'M68300' to 'M68008'
Changed all references from 'skeletonCodeDialog' to 'gui'
Made sure undo works with the new push and pop structure
Changed the display of the IR to 2 bytes
Introduced systemStack ops in illegal, link, pea, trap, trapv and unlk
Updated the Makefile to reflect some filename changes

PROBLEMS

* Did NOT introduce systemStack ops in MOVEM (which had a seperate block of code for each individual register...?!)
Non-specific stack operations that change a[7] screw up the systemStack

Edsko

1.111 Re: Final code changes/cleanups (sent by Diarmuid)

(Sent at Sun May 5 11:51:01 2002)

On Sun, May 05, 2002 at 10:34:34AM +0100, Edsko de Vries wrote:

I wrote that one when I was really tired. It is a matter of personal shame on my part but I would be interested to hear how you would suggest I approach it?

This one should be reasonably solveable but the project is wrapping up and I don't think that we want to fiddle too much at this point.

Diarmuid

1.112 Test mail (sent by Diarmuid)

(Sent at Sun May 5 12:12:01 2002)

Sun May 5 12:11:33 IST 2002

1.113 Test mail 2 (sent by Diarmuid)

(Sent at Sun May 5 12:15:01 2002)

Sun May 5 12:14:53 IST 2002

1.114 Test mail (sent by Diarmuid)

(Sent at Sun May 5 12:30:01 2002)

Sun May 5 12:28:59 IST 2002

1.115 Test mail 3 (sent by Diarmuid)

(Sent at Sun May 5 12:37:02 2002)

Sun May 5 12:35:57 IST 2002

1.116 Division of tasks (sent by Edsko)

(Sent at Sun May 5 20:24:01 2002)

Hi,

Declan has mailed us to say that we need to specify the division of tasks in the project report. I have set up a document describing the disision of tasks,

this can be found at

<http://matrix.netsoc.tcd.ie/edsko/68sim/tasks.pdf>

(The source can be found at the same location, tasks.tex). If you all agree with it, Alan, could you please include it in the project report. Feel free to change the wording by the way :)

Let me know if you've have any suggestions/add-ons.

Edsko

1.117 user manual attached (sent by Annie)

(Sent at Tue May 7 15:27:01 2002)

Hello,

Here is the user manual that myself and Laura have been working on in tex format. Let me know if you have any problems with it.

Annie

(Skipping MIME encoded file . . . Ok)

1.118 CLOSING EMAIL (sent by Edsko)

(Sent at Wed May 8 18:09:01 2002)

Hey,

This is the last official mail on this mailing list. Though the mailing list is still open, additional entries will not be included in the project documentation.

Thanks to all for a great project :)

Edsko