```
vim: textwidth=32 et sw=4:
```

To print these cards, use a2ps(1) with the following options:

```
a2ps -B -9 -r -f 12
```

Each card is about 32 characters wide.

asdlasm.nw

Implements the Asm3.assembler interface. Emits instructions as RTLs in ASDL syntax. Implemented for somebody who wanted it but never used it.

asm3.nw

Defines assembler interface.

Semantics need clarification. Easier handling of symbols desirable.

ast.nw (ast, ast.asdl, astpp)

Definition of abstract syntax of C-- in ASDL. Provides also pretty printer for AST.

ast3ir.nw

Translates AST to CFG/Assembly code. Main connection between front and back end.

Problems: translation of continuations, unsupported statements, spans. See qc--/doc/*.

astasm.nw

Implements Asm.assembler by translation of assembly code to AST, which then is pretty printed. Used mainly for debugging.

automaton2.nw

Interface for an automaton for resource allocation. Used to allocate slots in frame and for calling conventions.

Might be better to use OO style for interface definition.

auxfuns.nw

Auxiliaries: modules Option, Bij (Bijection).

Bijection probably unused.

backplane.nw

Defines Lua primitives to control the back end using Lua.

block.nw

Abstraction for recursively composable memory blocks.

cfg4.nw

Control−flow graph (CFG).

Problems: see qc−−/doc/*

cfgprop.nw

Mutable container for attachments to CFG nodes. Used for liveness information.

cfgutil.nw

Convenience functions for CFG traversal and manipulation.

colorgraph.nw

Graph−coloring register allocator with Lua interface.

const2.nw

Equation solver for RTL expressions.

contn.nw

Representation of continuations as pointer to pair of pointers.

copyinout.nw

Translation of parameter list into list of assign nodes in the CFG.

dotasm.nw

Implementation of Asm3.assembler. Emits CFG of procedure in DOT syntax.

driver2.nw

Exposes back end data structures to Lua.

Problems: too centralized. Better put Lua interfaces into modules that provide the O’Caml implementation.

dummyexpander.nw

OCamlBurg based code expander for imaginary Dummy target.

Lua interface.

elab.nw

Check of static semantics, build up Fenv.

Problems: complicated, many passes.

eqn.nw

Linear equation solver over integers. Used by const2.nw.

error.nw

Error handling operators. Mainly used by front end.

expander.nw

Toy code expander, not based on OCamlBurg.

expcheck.nw

Type checking routine for AST expression.

fenv.nw

Fat environment, symbol table.

Problems: Distinction between clean and dirty environment makes reuse of client code harder.

Symbols are back end specific and have leaked here into the front end.

idgen.nw

Generator for unique names.

impossible.nw

Exception for internal errors.

lifetime.nw

Intervals over abstract points. Used to represent lifetimes of registers in the linear scan register allocator.

Problems: unique definition of an empty interval difficult.

linscan.nw

Linear scan register allocator. Usually not used.

live.nw

Liveness analysis. Annotates nodes in a CFG with liveness informations.

luautil.nw

Defines a function that searches for a Lua file along a search path or directories. As a special case, a directory can be a Lua table. Used to execute Lua startup code for the compiler.

main2.nw   (qc--.lua, this.in, qc--.pod)

Entry point for compiler. Command line handling. Lua startup code, manual pages.

mangle.nw

Name mangling framework. Used by assemblers.

memalloc.nw

Incremental allocation of a Block.t memory
block using allocate(), align(). Used during translation of stack data.

minisparc.nw (msparcenc, msparcdec)

Constructor functions for SPARC instructions in RTL representation and matching decoder (recognizer). Code taken from machine generated code and manually adopted.

Problems: incomplete.

msparcasm.nw

Implementation of Asm3.assembler interface for SPARC architecture.

Problems: affected by unclear semantics of Asm3.assembler.

newer.nw

C program to compare file time stamps. Used in mkfile.

parser.nw (parse, scan)

Scanner and parser for C--.

placevar.nw

Replaces variables in RTLs with temporaries. Interface to Lua.

proc.nw

Data type definition for procedure representation in the back end.

register.nw

Definition of registers as well as maps and sets of registers.

Problem: Not part of the RTL framework and thus requires extra work to convert between registers and RTL locations.

rtlasdl.nw

Definition of ASDL syntax for RTLs. Used by asdlasm.nw.

rtldebug.nw

Type check for RTLs for debugging.

rtleval2.nw

Constant folding for RTL expressions.

rtlop.nw

Translation of C-- operators to RTL operators. Types of all RTL operators.

rtlutil.nw

Functions on RTLs: Width of an RTL, read and write sets, substitutions, evaluation time classification, translation to AST, translation to string.

rtlx.nw

Export of RTL in ASDL syntax to a file channel.

space.nw

Definition of an RTL space.

sparcasm.nw

Implementation of Asm3.assembler interface for the SPARC.

Problems: incomprehensible, unused. Replaced by msparcasm.

sparcexpander.nw

OCamlBurg code expander for the SPARC. Uses the constructor functions from minisparc.nw.

Problems: incomplete, no support for floating point operations and calls.

stack.nw

Contains no code, discusses stack layout. Probably outdated.

strutil.nw

String utilities: sets and maps over strings.

talloc.nw

Source for temporaries, i.e. registers.

Problems: the Simple module has no clients, implementation of Multiple seems too complicated. See comment at top of file.

target2.nw

Description of a target and its calling conventions.

Problems: Continuation representation is part of target description but should be part of calling convention,

targets.nw

Description of all known targets

topsort.nw

Topological sort routine. Used by elab.nw to sort declarations.

types.nw

Type language for C--.

base.nw

Type declarations for base types. Was introduced by Daniel Damien but is currently not thoroughly used.

bitops.nw

Operations on bit vectors.

bits.nw

Definition of a bit vector data type.

Problems: internal representation is fixed to 64 bits. No conversion to and from floating point representations.

reladdr.nw

Data type for relocatable addresses. No longer used.

reloc.nw

Data tyoe for relocatable addresses.

Problem: Sledlib defines another type for relocatable addresses. This one was modeled after the other. But do we need really two definitions?

rtl.nw

Definition of RTLs.

Problems: Should registers be part of an RTL? Separation between public and private representation is often broken by clients. Too simple representation of operators.

symbol.nw

Definition of an assembler symbol.

uint64.nw

Implementation of unsigned 64 bit integers in C with an OCaml binding. Currently not used.