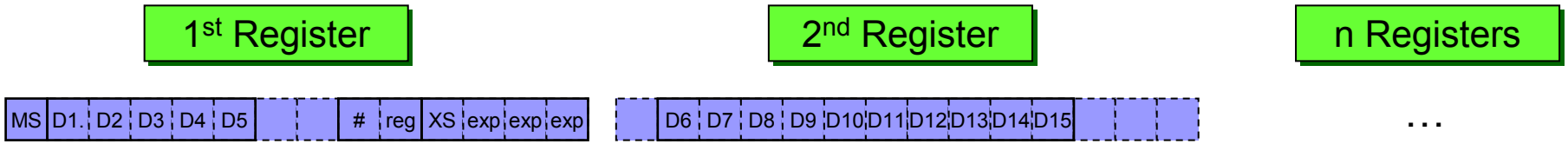
Mcode Multi-precision Library (MPL) For the HP41

- ✓ ~1000 digits, exponents up to ±2048
- ✓ Complete set of basic floating point arithmetic
  - +,-, *, /
- ✓ Some special floating point functions
  - x^2, y^x, 1/x
- ✓ (semi) Complete set of MPN-NN arithmetic functions
  - +, -, *, */10^x
- ✓ Complete set of multi-precision number (MPN) handling and I/O functions
  - Create, set, view, get MPN
  - Convert between MPN and Normal Number
  - Copy, CHS, CLX MPN
- ✓ Custom MPL specific error messages
- ✓ Entirely in MCODE (~4k, line-by-line documentation)
- ✓ Things still to do
  - sqrt(x), e^x, ln(x), Fact(x) => Open Source project for community?
  - Comprehensive overflow/underflow treatment
  - More debugging / user testing

MPNs Are Stored 10 Digits To A Register

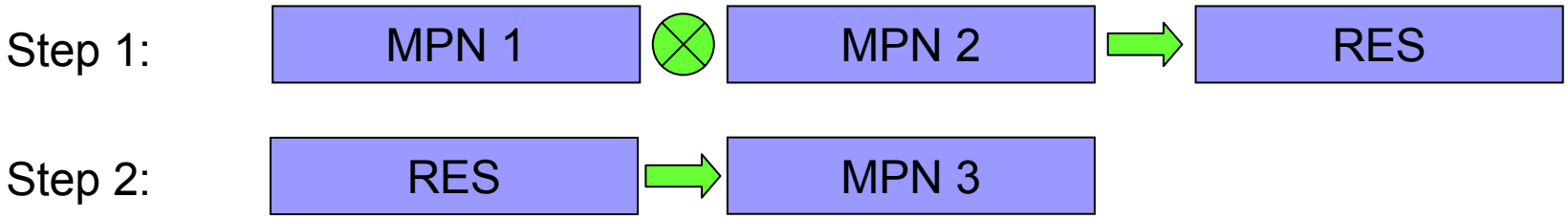


- 4 digits for exponent and exponent sign stored in first register
- Exponent stored in hex and 2-complement
- All calculations are done using either a full register per operation (+, -) or a 5-byte Nybble (*, /, 1/x, x^2, y^x)
- MPNs are identified by the user register number of the first register
- #reg – number of registers per MPN

⇨ Opportunity to increase digit density by

- ⇨ Storing 14 digits per register instead of 10 for a 40% gain in digit density with very limited additional overhead (calculations would be done on 7-byte nybbles with slightly more difficult carry handling
- ⇨ Storing digits in hex form for an additional ~15% gain, however with considerable additional overhead (conversion to and from)

Internal Calculations Use *RES* MPN Similar to HP71b



- *RES* is created as a buffer (ID=33) with the MPINIT command
  - One extra register (=2 nybbles) at end for higher rounding accuracy in last digit
- A second temporary *TMP* buffer (ID=44) is needed for 1/x,/,x^2,y^x
  - These buffers can be quickly created and deleted to free up memory, without affecting existing MPNs
- General template for MP operations modeled after normal RPN
  - Before operation
    - MPN1 in X-register
    - MPN2 in Y-register
    - MPN3 in L-register
  - After operation
    - MPN1 in L-Register
    - MPN3 in X-Register
    - MPN2 lost

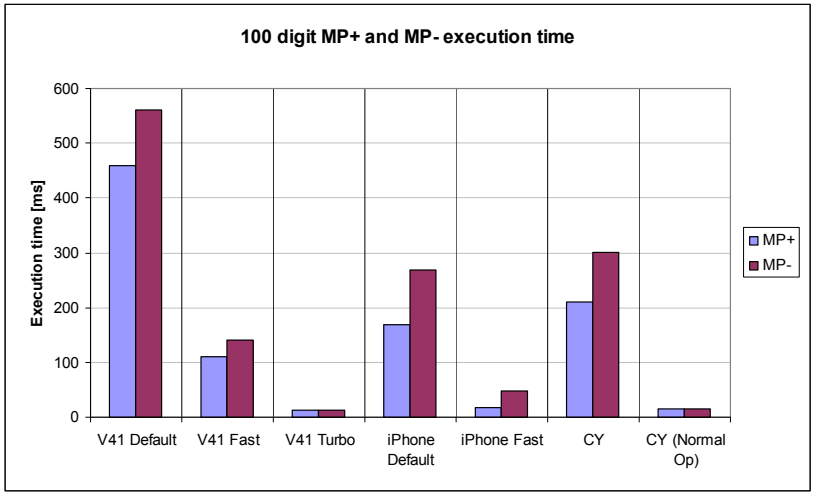
Operational Logic for Calculations

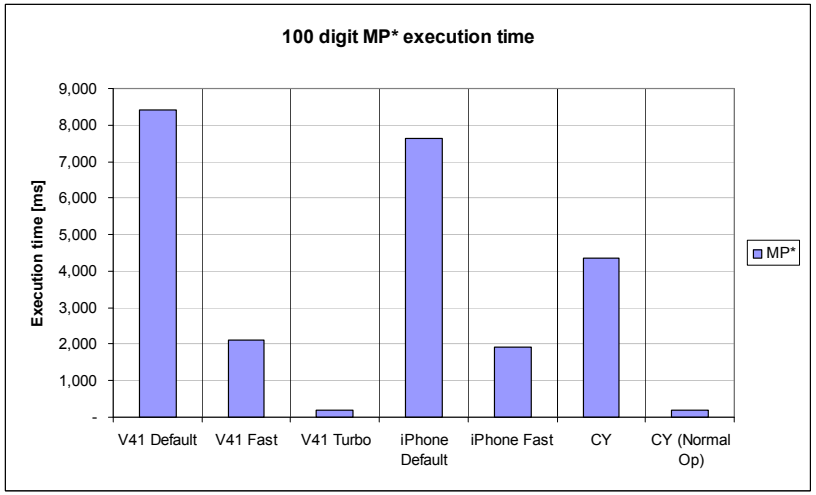
- Addition – MP+
  - Straight forward addition register by register with carry
- Subtraction – MP-(1)
  - Complement subtrahend
  - Addition one register at a time with carry
  - Uncomplement result if necessary
- Multiplication – MP*
  - MPN's are split into 5-digit nybbles
  - Nybble by nybble multiplication with on-the-fly carry calculations(2)
- Division – MP/
  - 1/x calculation via Newton iteration $X_n=X_{n-1}*(2-X*X_{n-1})$
  - X0 via high-precision 1/x calculation from OS (13 digit accuracy)
  - Rounding of last digit to ensure x*1/x = 1
  - Use of MP* routine for final result
- Squaring – MPX^2
  - Use of MP* routine, fed with both MPNs being x
- Exponentiation – MPY^X
  - Binary exponentiation using MP* and MPX^2 routines

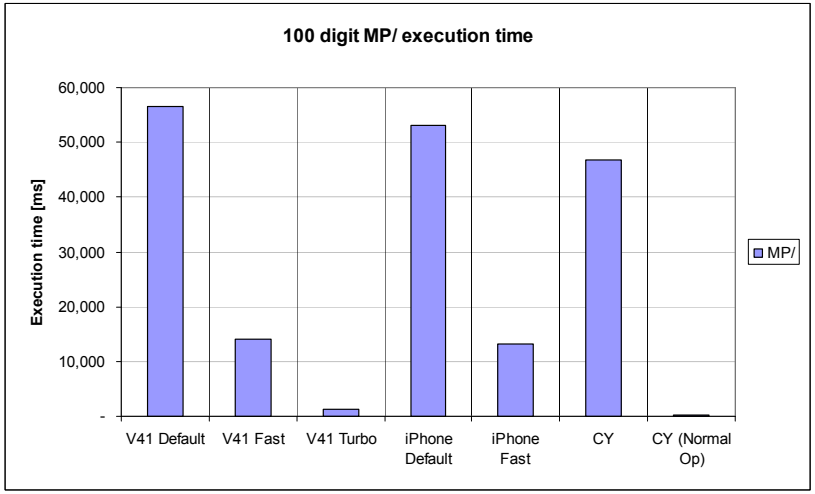
(1) Thanks to Egan Ford for invaluable tips and pointers
(2) Algorithm from Knuth – The Art of Computer Programming

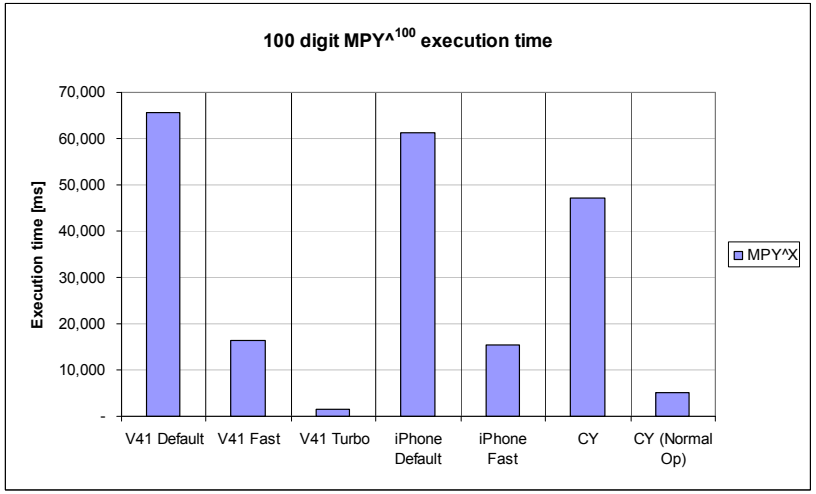
Long Run-Times But Bearable on Emulators

Skilled MCODE programmers can most likely reduce these times significantly









QRG For Functions and Error Messages

MPL Maintenance		Input	Output	Stack
MPINIT	Initialize the MPL. Creates two buffers RES and TMP with ID 33 and 44.	X - Number of registers per MPN	None	unaltered
MPUNL	frees up space by deleting the MPL buffers	none	None	unaltered
MPL?	Checks if the MPL is loaded. Skip/no if not, NoSkip/yes otherwise	none	None	unaltered
MPNOR?	Delivers to X the number of registers per MPN for the current MPL	none	X - #of reg per MPN	lifted
MPNOD?	Delivers to X the number of digits per MPN for the current MPL	none	X - # of dig per MPN	lifted

MPN Maintenance		Input	Output	Stack
MPNEW	Creates a new MPN, setting its exponent, sign, first 10 digits and starting reg. Each MPN is identified by its starting reg for all MPL functions	Z - exponent and sign btw+/- 2048 Y - sign and up to first 10 digits. The decimal is always after the first digit. No exponent X - first reg for MPN.	none	unaltered
MPSTO	Sets 10 digits at a time for an MPN. User is in charge of picking the right register. Leading 0s will be automatically padded.	Y - register to store digits into X - 10 digits to store	none	unaltered
MPGET	Get first 10 digits and exponent of MPN	X - MPN (first reg)	Y - exponent X - first 10 digits	lifted
MPGETM	Get first 10 digits of MPN	X - MPN (first reg)	X - first 10 digits	lifted
MPGETX	Get exponent of MPN	X - MPN (first reg)	X - exponent	lifted
MPVIEW	MPN-Viewer. View MPN(X) in X 10 digits at a time. R/S for next 10 digits, / for previous, <- to exit. 'RAD' signifies last reg	X - MPN(X)	none	unaltered

MP Arithmetic		Input	Output	Stack
MP+	Add two MPN's from X and Y and place result into MPN in L. MPN(L) can be the same as MPN(X) or MPN(Y)	Y - MPN(Y) X - MPN1(X) L - MPN(L) - Result	X - MPN(L) Result L - MPN(X)	dropped
MP-	Subtract MPN(X) from MPN(Y) and place result into MPN in L. MPN(L) can be the same as MPN(Y) but not as MPN(X)	Y - MPN(Y) X - MPN1(X) L - MPN(L) - Result	X - MPN(L) Result L - MPN(X)	dropped
MP*	Multiply MPN(X) by MPN(Y) and store result in MPN(L). MPN(L) can be the same as MPN(X) or MPN(Y)	Y - MPN(Y) X - MPN1(X) L - MPN(L) - Result	X - MPN(L) Result L - MPN(X)	dropped
MP/	Divide MPN(Y) by MPN(X) and store result in MPN(L). MPN(L) can ***not*** be the same as either MPN(X) or MPN(Y)	Y - MPN(Y) X - MPN1(X) L - MPN(L) - Result	X - MPN(L) Result L - MPN(X)	dropped
MP1/X	Calculate 1/MPN(X) and store in MPN(L). MPN(L) can not be the same as MPN(X)	X - MPN(X) L - MPN(L) - Result	X - MPN(L) Result L - MPN(X)	dropped
MPX^2	Calculate the square of MPN(X) and store in MPN(L). MPN(L) can be the same as MPN(X)	X - MPN(X) L - MPN(L) - Result	X - MPN(L) Result L - MPN(X)	dropped
MPY^X	Exponentiate MPN(Y) by the normal number NN(X) and store result in MPN(L). MPN(L) can not be the same as MPN(X)	Y - MPN(Y) X - NN(X) L - MPN Result	X - MPN Result L - NN(X)	dropped

MPN Manipulation		Input	Output	Stack
MPSEXP	Set the exponent of the existing MPN(Y). This can be used for fast multiplication or division by powers of 10	Y - MPN(Y) X - New exponent +/-2048	X - MPN(Y) L - New Exponent	dropped
MPCHS	Change the sign of MPN(X)	X - MPN(X)	none	unaltered
MPCLX	Clear the MPN(X) (i.e. +0.0 e0)	X - MPN(X)	none	unaltered
MPCPY	Copy existing MPN(Y) onto existing MPN(X), replacing its content	Y - Source MPN(Y) X - Target MPN(X)	X - Target MPN(X) L - Source MPN(X)	dropped

MPN - NN Interaction		Input	Output	Stack
MPM2N	Convert the MPN(X) into a normal number. Exponent is modulo 100	X - MPN(X)	X - Normal Number	
MPN2M	Convert a normal number NN(Y) into the existing MPN(X)	Y - Normal Number NN(Y) X - Target MPN(X)	none	unaltered
MPM+N	Add the normal number NN(X) to the MPN(Y) and store the result in MPN(L). MPN(L) can be MPN(Y)	Y - MPN(Y) X - normal number NN(X) L - MPN(L) - Result	X - MPN(L) L - NN(X)	dropped
MPM-N	Subtract the normal number NN(X) to the MPN(Y) and store the result in MPN(L). MPN(L) can be MPN(Y)	Y - MPN(Y) X - normal number NN(X) L - MPN(L) - Result	X - MPN(L) L - NN(X)	dropped
MPM*N	Multiply the MPN(Y) by the normal number NN(X) and store the result in MPN(L). MPN(L) can be the same as MPN(Y)	Y - MPN(Y) X - normal number NN(X) L - MPN(L) - Result	X - MPN(L) L - NN(X)	dropped

Comparison Functions		Input	Output	Stack
MPX=0?	Check if the MPN(X) is equal to 0 in both mantissa and exponent. Skip/No if not, otherwise No Skip/Yes	X - MPN(X)	none	unaltered
MPX=1?	Check if the MPN(X) is equal to 1 in the mantissa. The exponent is ignored. Skip/No if not, otherwise No Skip/Yes	X - MPN(X)	none	unaltered

System Functions - handle with care		Input	Output	Stack
MPLCNR	Change the Number of Registers for the current MPL	none	none	unalterd
MPNCNR	Change the Number of Registers for the current MPN	none	none	unalterd

	Error Msg	Functions	Explanation
MPL Level			
	MPN SIZE=0	MPINIT	The desired size in registers for a MPN is invalid. It needs to be 1<MPN Size<100. Please enter a valid size and retry.
	MPN SIZE>99	MPINIT	The desired size in registers for a MPN is invalid. It needs to be 1<MPN Size<100. Please enter a valid size and retry.
	MPN SIZE<1	MPINIT	The desired size in registers for a MPN is invalid. It needs to be 1<MPN Size<100. Please enter a valid size and retry.
	MPN SIZE<0	MPINIT	The desired size in registers for a MPN is invalid. It needs to be 1<MPN Size<100. Please enter a valid size and retry.
	NO MPL(RES)	All MP functions	No MPL loaded. Please initialize the MPL with MPINIT and a proper MPN size
	NO MPL(RES)	All MP functions	No MPL loaded. In particular the internal RES buffer was not found. Please initialize the MPL with MPINIT and a proper MPN size
	NO MPL(TMP1)	All MP functions	No MPL loaded. In particular the internal TMP buffer was not found. Please initialize the MPL with MPINIT and a proper MPN size
	MPN-MPL MISM	All MP functions	One or more of the selected MPNs use a different number of registers than the currently active MPL. Please unload (MPUNL) the current MPL and initialize a new one with the right number of registers per MPN
MPN Address - General			
	NO MPN	All MP functions	One or more of the selected MPNs are invalid. This happens most often by inadvertently using a register number which is not the first register of a MPN. Please check your MPN entries (including L) and retry
	INVALID MPN	All MP functions	One or more of the selected MPNs are invalid. This happens most often by inadvertently using a register number which is not the first register of a MPN. Please check your MPN entries (including L) and retry
	BAD MPN	All MP functions	One or more of the selected MPNs are invalid. This happens most often by inadvertently using a register number which is not the first register of a MPN. Please check your MPN entries (including L) and retry
MPN Address Specific			
	X-MPN INVALID	All MP functions	The number in x does not represent a valid MPN (e.g. negative, smaller than 0, etc)
	Y-MPN INVALID	All MP functions	The number in Y does not represent a valid MPN (e.g. negative, smaller than 0, etc)
	L-MPN INNVALID	All MP functions	The number in L does not represent a valid MPN (e.g. negative, smaller than 0, etc)
	X NO MPN	All MP functions	The number in X does not represent a MPN. Most likely it was an inintended input (e.g. switching X and Y regs)
	Y NO MPN	All MP functions	The number in X does not represent a MPN. Most likely it was an inintended input (e.g. switching X and Y regs)
	Z NO MPN	All MP functions	The number in X does not represent a MPN. Most likely it was an inintended input (e.g. switching X and Y regs)
MPN Memory Location			
	MPN TOO LONG	MPNEW	There is not enough room to store all registers for this MPN. Please check the current SIZE setting, start at a lower register number and try again
	X TOO LONG	All MP functions	There is not enough room to reach all registers for the MPN in X. Most likely you entered a wrong register number or changed the SIZE after enetering the MPN
	Y TOO LONG	All MP functions	There is not enough room to reach all registers for the MPN in Y. Most likely you entered a wrong register number or changed the SIZE after enetering the MPN
Exponent Values			
	BAD EXP	MPNEW, MPSEXP	The desired exponent is invalid (e.g. a number <1 or >2048, not integer)
	EXP INVALID	MPNEW, MPSEXP	The desired exponent is invalid (e.g. a number <1 or >2048, not integer)
	EXP >+/- 2048	MPNEW, MPSEXP	The desired exponent is > than +2048 or <-2048. Attention: Overflow/Underflow treatment not fully implemented yet!
MPN Values			
	MPN-VAL>1e10	MPNEW, MPN2M, MPSTO	The desired 10 digits for the MPN are > than 1e10 (i.e. they have more than 10 digits or an exponent
	MPN-VAL<0	MPNEW, MPN2M, MPSTO	The desired 10 digits for the MPN are < than 1 (i.e. they have a 0 as the only digit left of the comma)
	DIGITS>1e10	MPNEW, MPN2M, MPSTO	The desired 10 digits for the MPN are > than 1e10 (i.e. they have more than 10 digits or an exponent
	|Digits|<1	MPNEW, MPN2M, MPSTO	The desired 10 digits for the MPN are < than 1 (i.e. they have a 0 as the only digit left of the comma)
Special			
	BAD 1/X #>1	MP1/X, MP/	Error in the last digit treatment for 1/X. Please record your steps to reproduce this error and contact the author via the Forum.
	X0>10 IN 1/X	MP1/X, MP/	Error in the 2nd part of the 1/X iteration. Please record your steps to reproduce this error and contact the author via the Forum
	GetNyb Error	MP*	Error in loading a nybble in MP*. Please record your steps to reproduce this error and contact the author via the Forum
	FO IN C1stR	MP*	Odd nybble trying to load in first register Carry treatment in MP*. Please record your steps to reproduce this error and contact the author via the Forum