Minimal UART CPU Instruction Set

Legend: A=accumulator, R=result, M=most significant byte, ?=undefined, -=unchanged

Instruction			Description	Target	Operand		Accumulator	Flags			Clock
Name	DEC	HEX			Type	Size	Change	N	C	Z	Cycles
NOP	0	**00**	No Operation	none	none	0	none	-	-	-	16
WIN	1	**01**	Wait on Terminal Input	A	UART	0	result	A	0	A	>=10
OUT	2	**02**	Output to Terminal	UART	A	0	none	1	0	0	4
CLC	3	**03**	Clear Carry In Flag	none	none	0	none	1	0	0	5
SEC	4	**04**	Set Carry In Flag	none	none	0	none	0	1	1	5
LSL	5	**05**	Logical Shift Left (=ASL)	A	none	0	result	R	R	R	5
ROL	6	**06**	Rotate Shift Left	A	none	0	result	R	R	R	5
LSR	7	**07**	Logical Shift Right	A	none	0	result	R	R	R	13
ROR	8	**08**	Rotate Shift Right	A	none	0	result	R	R	R	12
ASR	9	**09**	Arithmetic Shift Right	A	none	0	result	R	R	R	15
INP	10	**0A**	Terminal Input (0xff: empty)	A	none	0	result	-	-	-	4
NEG	11	**0B**	Negate	A	none	0	result	A	?	A	6
INC	12	**0C**	Increment	A	none	0	result	R	R	R	5
DEC	13	**0D**	Decrement	A	none	0	result	R	R	R	5
LDI	14	**0E**	Load from	A	immediate	1	result	-	-	-	4
ADI	15	**0F**	Add	A	immediate	1	result	R	R	R	5
SBI	16	**10**	Subtract	A	immediate	1	result	R	R	R	5
CPI	17	**11**	Compare	A	immediate	1	none	R	R	R	5
ACI	18	**12**	Add with Carry In	A	immediate	1	result	R	R	R	5
SCI	19	**13**	Subtract with Carry In	A	immediate	1	result	R	R	R	5
JPA	20	**14**	Jump to	PC	abs address	2	none	-	-	-	6
LDA	21	**15**	Load from	A	abs address	2	result	-	-	-	7
STA	22	**16**	Store A to	byte @	abs address	2	none	-	-	-	8
ADA	23	**17**	Add	A	abs address	2	result	R	R	R	8
SBA	24	**18**	Subtract	A	abs address	2	result	R	R	R	8
CPA	25	**19**	Compare	A	abs address	2	none	R	R	R	8
ACA	26	**1A**	Add with Carry In	A	abs address	2	result	R	R	R	8
SCA	27	**1B**	Subtract with Carry In	A	abs address	2	result	R	R	R	8
JPR	28	**1C**	Jump to	PC	rel address	2	none	-	-	-	9
LDR	29	**1D**	Load from	A	rel address	2	result	-	-	-	10
STR	30	**1E**	Store A to	byte @	rel address	2	none	-	-	-	10
ADR	31	**1F**	Add	A	rel address	2	result	R	R	R	11
SBR	32	**20**	Subtract	A	rel address	2	result	R	R	R	11
CPR	33	**21**	Compare	A	rel address	2	none	R	R	R	11
ACR	34	**22**	Add with Carry In	A	rel address	2	result	R	R	R	11
SCR	35	**23**	Subtract with Carry In	A	rel address	2	result	R	R	R	11
CLB	36	**24**	Clear	byte @	abs address	2	like target	0	1	0	8
NEB	37	**25**	Negate	byte @	abs address	2	like target	A	?	A	10
INB	38	**26**	Increment	byte @	abs address	2	like target	R	R	R	10
DEB	39	**27**	Decrement	byte @	abs address	2	like target	R	R	R	10
ADB	40	**28**	Add	byte @	abs address	2	none	R	R	R	9
SBB	41	**29**	Subtract	byte @	abs address	2	none	R	R	R	10
ACB	42	**2A**	Add with Carry In	byte @	abs address	2	none	R	R	R	9
SCB	43	**2B**	Subtract with Carry In	byte @	abs address	2	none	R	R	R	10
CLW	44	**2C**	Clear	word @	abs address	2	none	0	1	0	10
NEW	45	**2D**	Negate	word @	abs address	2	not preserved	M	?	M	12
INW	46	**2E**	Increment	word @	abs address	2	not preserved	M	M	M	12
DEW	47	**2F**	Decrement	word @	abs address	2	not preserved	M	M	M	12
ADW	48	**30**	Add	word @	abs address	2	not preserved	M	M	M	11
SBW	49	**31**	Subtract	word @	abs address	2	not preserved	M	M	M	12
ACW	50	**32**	Add with Carry In	word @	abs address	2	not preserved	M	M	M	12
SCW	51	**33**	Subtract with Carry In	word @	abs address	2	not preserved	M	M	M	13
LDS	52	**34**	Load from Stack	A	offset	1	result	?	?	?	9
STS	53	**35**	Store A on Stack	stack	offset	1	none	?	?	?	16
PHS	54	**36**	Push on Stack	stack	none	0	none	?	?	?	12
PLS	55	**37**	Pull from Stack	A	none	0	result	?	?	?	10
JPS	56	**38**	Jump to Subroutine	PC	abs address	2	not preserved	?	?	?	16
RTS	57	**39**	Return from Subroutine	PC	none	0	not preserved	?	?	?	14
BNE	58	**3A**	Branch on Non-Zero	PC	abs address	2	none	-	-	-	5/6
BEQ	59	**3B**	Branch on Zero	PC	abs address	2	none	-	-	-	5/6
BCC	60	**3C**	Branch on Carry Clear	PC	abs address	2	none	-	-	-	5/6
BCS	61	**3D**	Branch on Carry Set	PC	abs address	2	none	-	-	-	5/6
BPL	62	**3E**	Branch on Plus	PC	abs address	2	none	-	-	-	5/6
BMI	63	**3F**	Branch on Minus	PC	abs address	2	none	-	-	-	5/6