

.NLIST SEQ
.REM 1

IDENTIFICATION

PRODUCT CODE: AC-E815B-MC
PRODUCT NAME: CKLPPFB0 LP20 MODULE
PRODUCT DATE: SEPTEMBER 1978
MAINTAINER: DEC/X11 SUPPORT GROUP

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITALS COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1976,1978 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

"LPPFB" IS AN I/O MODULE THAT EXERCISES THE LP20 CONTROLLER AND EITHER AN LP10 OR LP05 LINE PRINTER THAT MAY BE CONNECTED TO THE CONTROLLER.

2. REQUIREMENTS

HARDWARE: ONE LP20 LINE PRINTER CONTROL UNIT AND EITHER AN LP10 OR LP05 LINE PRINTER CONNECTED TO THE CONTROLLER

STORAGE: LPPFB REQUIRES:
1. DECIMAL WORDS: 819
2. OCTAL WORDS: 1463
3. OCTAL BYTES: 3146

3. PASS DEFINITION

A SINGLE PASS CONSISTS OF 32. ITERATIONS OF THE TEST SEQUENCES.

4. EXECUTION TIME

RUNNING ALONE ON A PDP11/40 USING AN LP10, THE INITIAL PASS (WITH PRINTING ENABLED) TAKES APPROXIMATELY 40. SECONDS. SUBSEQUENT PASSES (WITH PRINTING INHIBITED) TAKE APPROXIMATELY 10. SECONDS.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 175400 VECTOR: 754 BR1: 4 DEVCNT: 1

REQUIRED PARAMETERS:

"SR1" IS USED TO SPECIFY MODULE OPTIONS AS DESCRIBED BELOW:

SR1 (LO BYTE) BIT02 IS USED TO SPECIFY THE TYPE OF CHARACTER SET. IF IT IS A "1" IT INDICATES A 64. CHAR SET - ANY OTHER COMBINATION IN THE LO BYTE SIGNIFIES A 96. CHAR SET.

SR1 (HI BYTE) THE HIGH BYTE SPECIFIES PRINTING OPTIONS AS SHOWN BELOW:

SRI+1 = 0 PRINT ON FIRST PASS ONLY
SRI+1 = 1 NEVER PRINT
SRI+1 = 2 PRINT ON ALL PASSES

THE DEFAULT VALUE FOR SRI IS 000000, WHICH SIGNIFIES
64. CHAR SET AND PRINT ON FIRST PASS ONLY.

6. MODULE OPERATION

- A. SET UP INTERRUPT SERVICE VECTORS.
- B. TEST (RAM) MEMORY WITH DATA TEST.
- C. TEST (RAM) MEMORY ADDRESSING LOGIC.
- D. TEST PRINTING USING A TEST PATTERN OF ALL LEGAL CHARACTERS
THIS TEST WILL USE RAM TRANSLATION AND CAUSE INTERRUPTS
AT EACH CHARACTER.
- E. REPEAT STEPS C. THRU D. THREE (3) TIMES
THEN DO ENDPAS CALL.

7. OPERATION OPTIONS

NONE

!


```

286 ;RAM MEMORY BASIC ADDRESS TEST.
287 ;
288 ;TEST2: CLR R0 ;R0=CURRENT (RAM) ADDRESS
289
290 000506 005000
291 000510 012777 001400 177506 1$: MOV #LOINIT1RSTERR,@LPCSRA ;INIT LP20
292
293 000516 010077 177516
294 000522 010077 177510 MOV R0,@LPCCTR ;LOAD (RAM) ADDRESS.
295 MOV R0,@LPRAMD ;LOAD (RAM) DATA REGISTER.
296
297 000526 005200 INC R0 ;UPDATE (RAM) ADDRESS.
298
299 000530 020027 000377 CMP R0,#RAMTOP ;ALL (RAMS) LOADED?
300 000534 001765 1$ ;NOI
301
302 ;EACH (RAM) IS LOADED WITH ITS ADDRESS NOW TEST IT.
303 ;UPPER (RAM) ADDRESS.
304
305 000536 012701 000377 2$: MOV #RAMTOP,R1 ;R1=TOP OF RAM MEMORY.
306 000542 010167 177336 MOV R1,WASADR ;SAVE (RAM) ADDRESS FOR TYPEOUTS.
307
308 000546 010177 177466 3$: MOV R1,@LPCCTR ;LOAD (RAM) ADDRESS
309 000552 017767 177330 MOV @LPRAMD,AWAS ;ACTUAL RESULTS TO AWAS.
310
311 000560 010167 177322 MOV R1,ASB ;EXPECTED RESULTS TO ASB.
312 000564 042767 010000 BIC #PARITY,AWAS ;CLEAR PARITY BIT.
313
314 000572 026767 177310 177310 CMP ASB,AWAS ;TEST RESULTS.
315 000600 001410 4$ BEQ 4$ ;OK!
316
317 000602 017767 177416 177272 MOV @LPCSRA,ACSR
318 000610 016767 177410 177262 MOV @LPCSRA,CSRA
319 *****
320 DATA ERROR!!!
321 *****
322
323 000616 104404 000000
324
325 000622 005301 4$: DEC R1 ;ADJUST (RAM) ADDRESS.
326
327 000624 020127 000000 CMP R1,#RAMBOT ;ARE WE AT BOTTOM OF (RAMS)?
328 000630 001744 2$ BLOS 2$ ;NOI

```

```

326 ;TEST3 PRINTS A LINE OF ALL LEGAL CHARACTERS FOLLOWED BY A LINE OF ALL ILLEGAL CHARACTER
327 ;
328 ;TEST3:
329 000632 012777 001400 177364 RESTART: MOV #LOINIT1RSTERR,@LPCSRA ;INIT LP20.
330
331 000640 004767 000644 TEST3: JSR PC,RAMSET
332
333 000644 004767 001126 JSR PC,VFULOX ;LOAD VFU
334
335 ;START PRINTING
336
337 000650 012767 177753 000110 TEST3A: MOV #-21,,CYCCNT ;SET 21 LINES.
338
339 000656 012705 177574 1$: MOV #-132,,R5 ;SET CHARACTER COUNTER TO 132(10).
340 000662 012701 000770 MOV #BUFFER,R1 ;R1=TOP BUFFER ADDRESS.
341 000666 012702 000040 2$: MOV #40,R2 ;SET FIRST CHARACTER.
342
343 000672 110221 3$: MOVB R2,(R1)+ ;LOAD PRINT BUFFER.
344
345 000674 005205 INC R5 ;INCREMENT CHARACTER COUNT.
346 000676 001415 BEQ 5$ ;BRANCH IF LINE IS FINISHED.
347
348 000700 005202 INC R2 ;NEXT CHARACTER.
349
350 000702 132767 000002 177106 4$: BITB #2,SRI ;CHECK CHARACTER SET.
351 000710 001404 BEQ 4$ ;BRANCH IF 64 CHARACTER SET.
352
353 000712 022702 000200 5$: CMP #200,R2 ;LEGAL CHARACTER?
354 000716 001763 BEQ 2$ ;MAKE SPACE IF NOT LEGAL.
355 000720 000764 BR 3$ ;CONTINUE IF LEGAL.
356
357 000722 022702 000140 4$: CMP #140,R2 ;LEGAL CHARACTER?
358 000726 001754 BEQ 2$ ;MAKE SPACE IF NOT LEGAL.
359 000730 000760 BR 3$ ;CONTINUE IF LEGAL.
360
361 000732 112721 000012 5$: MOVB #12,(R1)+ ;ISSUE LINE FEED.
362 000736 105021 CLR B (R1)+ ;LOAD BUFFER TERMINATOR.
363
364 000740 012702 000770 MOV #BUFFER,R2
365 000744 004767 001240 JSR PC,LPTHE5 ;PRINT THE BUFFER.
366
367 000750 005267 000012 INC CYCCNT ;DONE ALL LINES?
368 000754 001340 BNE 1$ ;NO!
369
370 000756 104413 EPASS: ENDTTS,BEGIN ;SIGNAL END OF ITERATION.
371
372 000762 000167 177644 JMP TEST3 ;MONITOR SHALL TEST END OF PASS
373
374 000766 000300 CYCCNT: 0
375 000770 000214 BUFFER: -BLKW 140.
376
377 ;TEST LISTS
378
379 T02L: T02Z-T02L/2
380 10000
381 1

```



```

424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479

```

;THESE ROUTINES HANDLE PRINTING AND TEST PRINTING.
 ;DATBUF CONTAINS THE ACTUAL DATA AND IS 400(8) OR 256(10) WDS
 ;LONG. LOAD WITH CHRLOD. CHRLOD WILL PUT WHAT YOU TYPE INTO THE BUFFER.

```

DATSIZ: 0          ;+ # BYTES IN BUFFER
DATBFF: 0
TSTTYP: 0         ;TST TYPE TO BE RUN DURING TEST (RIGHT JUSTIF)
SETDEL: 0         ;I-SET UNDOCHR BEFORE SETTING GO,0=DO NOT
TWENT: 0          ;I-SET TWENT,0=DO NOT
PAGES: 0 10       ;SET PAG CNTR TO THIS BEFORE SETTING GO
LPSUM: 0          ;CHKSUM READ FROM LPT
GDSUM: 0          ;CHKSUM CALCULATED FROM GOOD DATA

```

;THIS ROUTINE ACTUALLY PRINTS ON THE LPT

```

LPTPNT: MOV #LOINTT,INTERR,@LPCSR A ;INIT LPT2.
        MOV #DATBUF,@LPCSR A ;SET UP BUS ADR
        MOV #DATSIZ,@LPCSR A ;SET UP + NR BYTES
        NEG @LPCSR A ;NEGATE THE COUNT
        MOV #PAGES,@LPCSR A ;SET UP PAGE COUNTER
        MOVB #TSTTYP,@LPCSR B ;SET UP TSTTYP
        JSR PC,CKSR ;GO CHECK "SR" OPTIONS
        BIS #103,@LPCSR A ;NOW SET GO AND PARENB
        EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

```

;-----

```

LPRINT: ;-----
        ;IRQS,BEGIN,IS ; QUEUE UP TO CONTINUE AT IS AND RTI
        BIT #100000,@LPCSR A ;ERROR FLAG SET?
        BEQ 25 ;NOI
        ENDS,BEGIN ;YES.
        BIS #200,@LPCSR A ; ;WAIT FOR A FLAG
        BNE 55 ;DONE

```

```

        MOV @LPCSR A,ACSR
        MOV @LPCSR A,CSRA
        MOV @LPCSR B,ASTAT

```

;*****

```

ORDERS,BEGIN,NULL ;
;*****
ENDS,BEGIN ;

```

```

5S: JSR PC,TSTFLG ;DECIDE WHAT TO DO
    RTS PC

```

;NOW FOLLOWS SUBR TO FIGURE OUT WHAT TO DO WHEN WE GET A
 ;FLAG WHILE WAITING FOR DONE.

```

TSTFLG: TSTB @LPCSR A ;IS IT DONE?
        ORL  @LPCSR A ;YES,RTN
        BIT  #40000,@LPCSR A ;IS IT PAGE ZERO?

```

```

480
481
482
483

```

```

SPFLG: MOV SPFLG ;NO
        MOV PAGES,@LPCSR A ;YES,RESET # PAGES
        ORL  #100,@LPCSR A ;SET GO AGAIN
        RTS PC ;CONTINUE

```

```

;HERE IS THE ROUTINE TO LOAD THE VFU. SET UP THE BUFFER
;AT "VFUDAT" TO THE DESIRED DATA AND "VFUSIZ" TO THE
;DESIRED SIZE AND RUN VFULOX.
484
485
486
487
488 001776* 032777 004000 176222 VFULOX: BIT #OPTOUT,@LPCSRB ;OPTICAL VFU?
489 002004* 001077 BNE 35 ;YES!
490
491 002006* 012777 001400 176210 MOV #LOINIT1RSTERR,@LPCSRRA ;INIT LP20.
492 002014* 012700 002740 MOV #VFUDAT,R0 ;CALCULATE END OF BUFFER
493 002020* 066700 000162 ADD #VFUSIZ,R2 ;POINTS TO NEXT BYTE
494 002030* 012777 002737 MOV #37,(R0) ;STOP VFU LOAD CODE
495 002040* 012777 002737 MOV #VFUDAT-1,@LPCRSAD ;SET UP BUS ADDR
496 002036* 016777 000144 MOV #VFUSIZ,@LPCPTR ;SET UP # NR BYTES
497 002044* 066777 000003 ADD #2,@LPCPTR ;2 EXTRA BYTES ON VFU LOAD
498 002052* 005477 176154 NEG @LPCPTR ;NEGATE THE COUNT
499 002056* 016777 177470 MOVB #TSTTYP,@LPCSRB ;SET UP TSTTYP
500 002064* 012777 000010 MOVB #10,@LPCSRRA ;SET UP FOR VFU LOAD MODE
501 002072* 052777 000003 BIS #3,@LPCSRRA ;NOW SET GO AND PARENB
502
503 002100* 032777 100000 176116 1S: BIT #100000,@LPCSRRA ;ERROR FLAG SET?
504 002106* 001402 BEQ 55 ;NO!
505
506 002110* 104410 000000* ENDS,BEGIN ;
507
508 002114* 032777 060200 176102 5S: BIT #60200,@LPCSRRA ;WAIT FOR A FLAG
509 002124* 004766 BEQ 51 ;PAGEZRO,UNDCR,DONE
510 002130* 057777 177612 JSR PC,TSTPLG ;DECIDE WHAT TO DO
511 002134* 057777 176070 TSTB @LPCSRRA ;TEST FOR DONE
512 002134* 000361 BPL 1S ;NOT DONE YET
513 002136* 007767 176100 MOVB @PTDAT,LPSUM ;SAVE THE LP CHKSUM
514 002144* 005067 177414 CLR GDSUM ;NOW CALCULATE GOOD SUM
515 002150* 012701 002740 MOV #VFUDAT,R1 ;CALCULATE ADR OF LAST BYTE
516 002154* 066701 000026 ADD #VFUSIZ,R1
517 002160* 012700 002737 MOV #VFUDAT-1,R0 ;FIRST BYTE IS START CODE
518 002164* 012700 002737 MOVB (R0),R2
519 002166* 060267 177372 ADD R2,GDSUM
520 002172* 020001 CMP R2,R1
521 002174* 001373 BNE 2S
522 002204* 006207 177400 177360 3S: RTS #77400,GDSUM ;LOOP BACK FOR REST OF BUFR
523
524
525 002206* 000204 VFUSIZ: 204 ;#BYTES OF DATA FOR DAVFU
526 002214* 012705 CLS LPTMES: 5 ;CLEAR TALLY.
527 002214* 012705 MOV #DATABUF,R5 ;DATABUF,R5
528
529 002220* 012225 1S: MOVB (R2)+,(R5)+ ;LOAD PRINTER BUFFER.
530 002222* 005267 000056 TALLY ;UPDATE TALLY.
531
532 002226* 057112 TSTB (R2) ;END OF MESSAGE?
533 002230* 001373 BNE 1S ;NO!
534
535 002232* 016767 000046 MOV TALLY,DATSIZ ;YES!
536 002240* 004767 177322 JSR PC,LPTPNT ;SET MESSAGE SIZE.
537 002244* 000207 RTS PC
538

```

```

;THIS ROUTINE IS CALLED TO TEST THE SOFTWARE REGISTER IN THE MODULE
;INTERFACE AND DECIDE HOW TO HANDLE PRINTING. THE LOCATION TESTED IS
;"SR1+1:" AND IS TESTED FOR THE FOLLOWING CODES:
539
540 ;SR1+1 = 0 PRINT ON THE FIRST PASS ONLY
541 ;SR1+1 = 1 NEVER PRINT - TEST MODE ONLY
542 ;SR1+1 = 2 PRINT CONTINUOUSLY
543
544 ;IF "SR1+1:" CONTAINS ANY OTHER CODE THE PROGRAM DEFAULTS TO
545 ;A CODE OF "000".
546
547 002246* 122767 000002 175543 CKSR: CMPB #2,SR1+1 ;PRINT CONTINUOUSLY ??
548 002254* 001412 BEQ 2S ;BR IF YES
549
550 002256* 122767 000001 175533 CMPB #1,SR1+1 ;NEVER PRINT ??
551 002264* 001403 BEQ 1S ;BR IF YES
552
553 002266* 005767 175542 TST PASCNT ;INITIAL PASS ??
554 002272* 001403 BEQ 2S ;BR IF YES
555
556 002274* 052777 000004 175722 1S: BIS #TSTMOD,@LPCSRRA ;SET TEST MODE - INHIBIT PRINT
557 002302* 000207 2S: RTS PC ;RETURN TO CALLER
558
559 002304* 000000 TALLY: 0
560

```

```

XLPFB0.P11 12-OCT-78 11:59
002306* 000214
564 002736* 000 356
565 002740* 000 356
566 002744* 000 356
567 002748* 077 077 020
568 002752* 002 024
569 002756* 000 000
570 002760* 000 000
571 002764* 000 000
572 002768* 006 006
573 002772* 000 000
574 002776* 000 000
575 002780* 006 006
576 002784* 000 000
577 002788* 000 000
578 002792* 000 000
579 002796* 000 000
580 002800* 000 000
581 002804* 000 000
582 002808* 000 000
583 002812* 000 000
584 002816* 000 000
585 002820* 000 000
586 002824* 000 000
587 002828* 000 000
588 002832* 000 000
589 002836* 000 000
590 002840* 000 000
591 002844* 000 000
592 002848* 000 000
593 002852* 000 000
594 002856* 000 000
595 002860* 000 000
596 002864* 000 000
597 002868* 000 000
598 002872* 000 000
599 002876* 000 000
600 002880* 000 000
601 002884* 000 000
602 002888* 000 000
603 002892* 000 000
604 002896* 000 000
605 002900* 000 000
606 002904* 000 000
607 002908* 000 000
608 002912* 000 000
609 002916* 000 000
610 002920* 000 000
611 002924* 000 000
612 002928* 000 000
613 002932* 000 000
614 002936* 000 000
615 002940* 000 000
616 002944* 000 000
617 002948* 000 000
618 002952* 000 000
619 000001 .END
    
```

```

XLPFB0.P11 12-OCT-78 11:59
CROSS REFERENCE TABLE -- USER SYMROLS
ACSR 000102R 171# 273* 461*
ADDR 000006R 137# 227
ADDR22= 001000R 189#
ASR 000106R 175# 257* 263 270 310* 313
ASRAT 000108R 176# 463* 268*
AMAS 000110R 176# 270 308* 311* 313
BEGTN 000000R 144# 276 319 371 448 452 457 466 468 506
BIT0 000001R 189#
BIT1 000002R 189#
BIT10 002000R 189#
BIT11 004000R 189#
BIT12 008000R 189#
BIT13 012000R 189#
BIT14 016000R 189#
BIT15 020000R 189#
BIT16 024000R 189#
BIT17 028000R 189#
BIT18 032000R 189#
BIT19 036000R 189#
BREAKS 00012R 189#
BR1 00012R 246
BR2 000013R 140#
BRODS 104441R 189#
BUFFER 000770R 341# 365 375#
BUSAL6 000020R 217#
BUSAL7 000040R 216#
CDATAS 104412R 189#
CKSR 002246R 446# 550#
CNFIG 000056R 159#
CSRA 000100R 129# 274* 317* 462*
CVCCNT 000766R 338# 368* 374#
DATBFF 001550R 430#
DATBUF 004246R 448# 527 564#
DATERS 104404R 189# 276 319
DATSTZ 001546R 429# 442 536*
DELPLD 001070R 211#
DNPLG 001074R 478# 483#
DONE 000200R 214#
DVID1 000014R 141#
END1S 104443R 189#
ERRTP 000000R 370#
ERRTP 000000R 370#
ERRTP 000000R 370#
GDSUM 001564R 436# 448 514* 519* 522*
GTPAS 104415R 189#
GD 000001R 120#
GWBUPS 104414R 189#
HRDCNT 000444R 189#
HRDRS 104405R 466
    
```


. = 003146R 375# 564#

. ABS. 000000 000
003146 001

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

XLPPB0 XLPPB0/SOL/CRF:SYM=DDXCOM,XLPPB0
RUN-TIME: 1 1 3 SECONDS
RUN-TIME RATIO: 173=4.6
CORE USED: 7K (13 PAGES)