Init after ATN, clear DAV, EOI, DATA

```
ATNIRQ:
   LDX #$FF                ; CLEAR STACK
   STX cmdnum
   TXS
   LDA io_ifr?             ; CLEAR IRQ FLAG
   LDA #$18                ; DAVO+EOIO
   ORA io_ieee_ctl         ; FREE CONTROL LINES
   STA io_ieee_ctl
   LDA #$FF
   STA io_ieee_data_out    ; FREE DATA LINES
```

Set ATNA, NRFD HI, NDAC LO

```
ATN10:
   LDA #$7                 ; DACO+RFDO+ATNA
   ORA io_ieee_ctl
   STA io_ieee_ctl
```

Wait DAV low or ATN HI

```
ATN20:
   BIT io_ieee_ctl
   BVC ATN30               ; DAV LO -> ATN30
   BMI ATN20               ; ATN LO -> ATN20 (wait loop)
   BPL ATN50               ; ATN HI -> ATN50
```

DAV LO — NRFD LO, Read data, NDAC HI

```
ATN30:
   LDA #$FB                ; NRFD LO
   AND io_ieee_ctl
   STA io_ieee_ctl
   AND #$20                ; EOIFLG
   STA eoiflg              ; SAVE EOI
   LDA io_ieee_data_in
   EOR #$FF                ; SAVE COMMAND
   STA data                ; NDAC HI
   LDA #$FD
   AND io_ieee_ctl
   STA io_ieee_ctl
```

Switch on received ATN command

```
DCDE:
   LDY #$0
   LDA data
   AND #$60
   CMP #$40                ; TALK?
   BEQ DCDE60
   CMP #$20                ; LISTEN?
   BEQ DCDE20
   CMP #$60                ; SECONDARY?
   BEQ DCDE70
   BNE DCDE80              ; OTHER
```

Process LISTEN address

```
DCDE20:
   LDA data
   CMP lsnadr              ; MY LISTEN ADDRESS
   BEQ DCDE40              ; UNLSN
   BNE DCDE30
   STY lsnact
DCDE30:
   STY lsntlk?             ; ADRSED - NOT PRIMARY ADDRSED
   JMP DCDE80

DCDE40:
   STA lsnact
   STY tlkact
```

Process TALK address

```
DCDE60:
   STY tlkact
   LDA data
   CMP tlkadr
   BNE DCDE30
   STA tlkact
   STY lsnact
   BEQ DCDE50              ; always
```

Process SECONDARY address

```
DCDE70:
   LDA lsntlk?             ; ADRSED
   BEQ DCDE80              ; NOT ADDRESSED
   LDA data
   STA orgsa
   PHA
   AND #$F
   STA sa
   PLA
   AND #$F0                ; CLOSE?
   CMP #$ED
   BNE DCDE80
   JSR close
```

End TALK/LISTEN address

```
DCDE50:
   LDA #$20
   STA sa                  ; DEFAULT SA
   STA orgsa
   STA lsntlk?             ; ADRSED - PRIMARY ADDRESSED
   BNE DCDE80              ; always
```

Wait end of DAV

```
DCDE80:
ATN40:
   BIT io_ieee_ctl
   BVC ATN40
   JMP ATN10
```

ATN HI — Check LISTEN

```
ATN50:
   LDA lsnact
   BEQ ATN60               ; (to talk)

   ; Set ATNA and set NRFD LO
   LDA #$FA                ; 255-RFDO-ATNA
   AND io_ieee_ctl
   STA io_ieee_ctl
   CLI
   JSR LISTEN
   JMP idle
```

LISTEN loop

```
LISTEN:
   ; NRFD HI
   LDA #$4                 ; RFDO
   ORA io_ieee_ctl
   STA io_ieee_ctl

   ; Wait DAV LO — ATN exits to IRQ
LSN10:
   BIT io_ieee_ctl         ; DAV: LO
   BVS LSN10

   ; Find channel
   JSR fndwch              ; WAS LDX SA
   BCS LSN15
   LDA chnrdy,X
   ROR A                   ; OK. OPEN FOR LISTEN
   BCS LSN30
LSN15:
   LDA orgsa               ; WAS TXA
   AND #$F0                ; SA=OPEN?
   CMP #$F0
   BEQ LSN30
LSN20:
   LDA sa
   CMP #$1
   BEQ LSN25
```

Ignore rest of data

```
   ; Wait for DAV hi
LSN21:
   BIT io_ieee_ctl
   BVC LSN21               ; LSN21
   ; NDAC HI
   LDA #$FD                ; 255-DACO
   AND io_ieee_ctl
   STA io_ieee_ctl
   ; return from Listen (timeout in host)
   RTS
```

Save (secaddr == 1)

```
   ; read and drop data if channel not open
LSN25:
   ; NRFD LO
   LDA #$FB                ; 255-RDFO - ACCEPT ALL DATA
   AND io_ieee_ctl
   STA io_ieee_ctl
   ; NDAC HI
   LDA #$FD                ; 255-DACO
   AND io_ieee_ctl
   STA io_ieee_ctl

   ; Wait for DAV HI
LSN26:
   BIT io_ieee_ctl         ; DAV HI
   BVC LSN26
   ; NDAC LO
   LDA #$2                 ; DACO
   ORA io_ieee_ctl         ; DAC LO
   STA io_ieee_ctl
   ; NRFD HI
   LDA #$4                 ; RFD HI
   ORA io_ieee_ctl
   STA io_ieee_ctl
   ; Wait for DAV HI
LSN28:
   BIT io_ieee_ctl         ; WAIT DAV LO
   BVC LSN28
   JMP LSN25               ; DO UNTIL ATN PULLED
```

Found open channel

```
LSN30:
   ; NRFD LO
   LDA #$FB                ; 255-RFDO
   AND io_ieee_ctl
   STA io_ieee_ctl
   ; store data
   AND #$20                ; EOI
   STA eoiflg
   LDA io_ieee_data_in
   EOR #$FF
   STA data
   SEI
   ; NDAC HI
   LDA #$FD                ; 255-DACO
   AND io_ieee_ctl
   STA io_ieee_ctl
LSN40:
   ; Wait DAV HI
   BIT io_ieee_ctl
   BVC LSN40
   ; NDAC Lo
   LDA #$2                 ; DACO
   ORA io_ieee_ctl
   STA io_ieee_ctl
   JSR put
LSTRTN:
   CLI
   JMP LISTEN
```

ATN HI — Check TALK

```
ATN60:
   ; Set ATNA and set NDAC HI
   LDA #$FC                ; 255-ANTA-DACO
   AND io_ieee_ctl
   STA io_ieee_ctl
   LDA tlkact
   BEQ ATN70
   CLI
   JSR TALK
   .atnl10:
ANT70
   JMP idle
```

TALK loop

```
TALK:
   JSR fndrch
   BCS NOTLK                           ; TEST IF CHANNEL READY
TALK1:
   LDX lindx
   LDA chnrdy,X
   BMI tlk05               ; TLK10
NOTLK:
   RTS

TLK10:
   ; Wait NRFD HI
   BIT io_dskcnt           ; RFD: HI
   BPL TLK10
   ; Put data on bus
   LDA chndat,X
   EOR #$FF
   STA io_ieee_data_out
   ; put EOI on bus, set DAV
   LDA chnrdy,X
   ORA #$E7                ; 255-EOIO-DAVO
   AND io_ieee_ctl
   STA io_ieee_ctl

   ; Wait NRFD LO, bus error on NDAC HI
TLK20:
   BIT io_dskcnt
   BPL TLK30               ; NRFD is Low — end
   BVC TLK20               ; NDAC is Low — loop
   ; end of transmission
   ; DAV & EOI HI
   LDA #$18                ; DAVO+EOIO
   ORA io_ieee_ctl
   STA io_ieee_ctl
   JMP idle

   ; get next data
TLK30:
   JSR LDA84               ; GET
   ; Wait for NDAC HI
TLK35:
   BIT io_dskcnt
   BVC TLK35
   ; Clear bus
   LDA #$FF
   STA io_ieee_data_out
   ; Clear DAV & EOI
   LDA #$18                ; DAVO+EOIO
   ORA io_ieee_ctl
   STA io_ieee_ctl
   ; Wait NDAC Low
TLK40:
   BIT io_dskcnt
   BVS TLK40
   BVC TALK1               ; always
```