Proprietary - Internal use only

Security Analysis report

SGS Thomson ST16CF54

Total CA system security is a function of several factors, including the system concept, its implementation in software, and the security of the hardware platform in which the hacker – accessible software resides.

The total security is as weak as its weakest link, and quite often that is the hardware platform. For this reason we examined the ST16CF54, advertised as a "safeguarded" smartcard processor, with "Very high security features" [Ref. 1], with the intention of verifying the manufacturer's security claims.

The analysis was done on ST16CF54A microprocessors, using commercially-available Kudelski / NagraVision cards, purchased with EchoStar (Dish Network) IRDs. The cards have not been used for normal viewing, so they do not contain data transmitted over the air. The card version is identified internally as "Rev309" (starting at memory location E032)

Please note that the ST16CF54A is offered for sale to commercial users with few hardware customization options. Breaking one ST16CF54A application will therefore provide a prospective hacker with valuable information concerning other applications.

The following discussion summarizes the analysis results, pointing out weak points and potential entries into the ST16CF54A. These weak points reflect on the security of commercial users of the ST16CF54A.

1. General.

The first line of protection of the ST16CF54A is its top metal mesh (grid) – see Fig. 1. The mesh protects the active parts of the IC from analytic probing using mechanical needles. At first looks it seems like a formidable barrier.

However, it is possible to penetrate this barrier using one of the well–known techniques described below. Once it is penetrated, we estimate that the security of the ST16CF54A can be broken quite easily.

The report describes:

- Mesh penetration techniques,
- Below – mesh security deficiencies,
- Possible cloning approaches.

1. Decapsulation and Mesh penetration techniques

The ST16CF54A is encapsulated in a standard protective material. Immersion in Red Fuming Nitric Acid (RFNA) and washing will leave a bare, clean die. Mounting and bonding the die in a ceramic package produces a working microprocessor in a convenient open package.

Proprietary - Internal use only

Security Analysis report

SGS Thomson ST16CF54

Total CA system security is a function of several factors, including the system concept, its implementation in software, and the security of the hardware platform in which the hacker – accessible software resides.

The total security is as weak as its weakest link, and quite often that is the hardware platform. For this reason we examined the ST16CF54, advertised as a "safeguarded" smartcard processor, with "Very high security features" [Ref. 1], with the intention of verifying the manufacturer's security claims.

The analysis was done on ST16CF54A microprocessors, using commercially-available Kudelski / NagraVision cards, purchased with EchoStar (Dish Network) IRDs. The cards have not been used for normal viewing, so they do not contain data transmitted over the air. The card version is identified internally as "Rev309" (starting at memory location E032)

Please note that the ST16CF54A is offered for sale to commercial users with few hardware customization options. Breaking one ST16CF54A application will therefore provide a prospective hacker with valuable information concerning other applications.

The following discussion summarizes the analysis results, pointing out weak points and potential entries into the ST16CF54A. These weak points reflect on the security of commercial users of the ST16CF54A.

1. General.

The first line of protection of the ST16CF54A is its top metal mesh (grid) – see Fig. 1. The mesh protects the active parts of the IC from analytic probing using mechanical needles. At first looks it seems like a formidable barrier.

However, it is possible to penetrate this barrier using one of the well-known techniques described below. Once it is penetrated, we estimate that the security of the ST16CF54A can be broken quite easily.

The report describes:

- Mesh penetration techniques,
- Below – mesh security deficiencies,
- Possible cloning approaches.

1. Decapsulation and Mesh penetration techniques

The ST16CF54A is encapsulated in a standard protective material. Immersion in Red Fuming Nitric Acid (RFNA) and washing will leave a bare, clean die. Mounting and bonding the die in a ceramic package produces a working microprocessor in a convenient open package.

EXHIBIT
103
No. SA CV 03-950 C 02-1178
DOC (JTLx)

The protective mesh feature of the ST16CF54A can be seen in Fig. 1. The mesh is metallic (AL/Ti). Each "cell" side is approx. 3.2 microns, giving a mesh opening of approx. 1.6 microns each side.

There are at least two methods by which such a mesh could be penetrated, to enable probing of selected signal lines:

- Photochemical Etching – Using the procedures outlined in [Ref. 2] one can open a local "window" in the protective mesh. The procedure involves coating the die with photoresist, exposing the window area, and using either wet chemicals or a small plasma etcher to clear a window through the top metal layer.

  It should be noted that [Ref. 2] is quite old, and uses the UV component of an optical microscope field illuminator to expose the resist. Considerable improvement can be achieved now by using a UV laser instead. This results in a much more precise and controlled exposure.

- FIB processing – A Focussed Ion Beam (FIB) system can be used to create probe-points at the desired location. The FIB simply mills a precise hole through the mesh, reaching down to the signal lines. The hole may be used for reaching down with a probing needle. As an alternative, a much smaller hole may be drilled through the mesh opening, and a conductive "antenna" deposited into the hole. The antenna makes contact with the signal line at the bottom, terminating with a metallic pad at the top. The pad is then used as a convenient probing point. (Fig. 2)

1. Below – mesh security deficiencies,

At the micro level, this structure seems very similar to earlier SGS Thomson microprocessors, so that knowledge gained from examination of other Thomson processors can be directly applied. Some of these structures are virtually identical, so that analysis could actually be carried out using a layer photo from one IC with another layer photo from another IC of the same family. (Fig. 3 a, b, c[1])

As a result of that, some vulnerable points could be easily located and verified – for example, trying out an a method of attack on a meshless processor prior to mesh penetration on a ST16CF54A. Finding the exact location of such these points on the ST16CF54A is done on a non-operating example of the ST16CF54A (Fig. 3c) which has undergone complete chemical stripping of the mesh. The points may also be seen through the mesh (Fig. 3b). Structure analysis is facilitated by the fact that the ST16CF54A is fabricated with a single metal signal level and a single poly layer.

The same single metal signal layer design is also responsible for the ease of probing the signal lines, compared with the more common 2-level microprocessor designs.

---

[1] Fig. 3a shows a portion of the instruction latch in another (meshless) SGS-Thomson device. Fig. 3b shows the same point through the mesh of a ST16CF54A. Fig. 3c shows the same point in the Poly layer of a stripped – down ST16CF54A. The complete layer picture can be assembled from these three images.

Specific points of vulnerability are:

1. **Data bus:** The data bus is going between all the ST16CF54A's major elements. Memory decryption (if any) is performed ahead of the bus, so that the bus may be probed in the clear at any of a large number of points.
   The bus lines are shuffled, but this shuffling may be found out either through circuit analysis or through logical analysis of the probed bus data during normal running (loops, I/O etc.).
   The bus includes "tails", or non-functional length at the end of a line (Fig. 4). Such "tails" facilitate mechanical probing, by reducing the risk of damage to the microprocessor by accidental line cuts with the probe needle.

2. **Program Counter:** A variation of the "single-needle" data extraction technique described in Reference [3] can be easily employed on the ST16CF54A. We located the instruction handler in its expected place near the microcode ROM. Quick analysis found several points through which the program counter can be made to "single step" through all memory – one is shown in Fig. 3. Thus, only one data needle is required to read all memory in eight "passes" over the data bus lines. The ST16CF54A does not have any protection mechanisms against it.

   The basic technique is widely known, and was made public in 1996 [Ref. 3]. It is a very potent tool, since all the memory (including portions protected by the security devices employed in the ST16CF54A) is revealed, ready for disassembly. Disassembly software for the 6805-family microprocessors is commercially available. Using such software (with addresses obtained through the running code) it is pretty straightforward to analyze the code.
   Several examples of code and data extracted from our sample ST16CF54A (EchoStar / NagraVision implementation) are shown in Appendix A.

3. **Memory Access Control Matrix:** Ref. [1] shows this matrix as controlling access from any of the six memory regions to any other of the six. Obviously, this should be a 6x6 matrix, with one side (either column or row) going to the "chip select" signals of the 6 memory regions, and the other side (either row or column) going through latches to the same chip selects.
   Such a structure was indeed found (Fig. 5). Its output line can be easily disabled, completely bypassing the intended protection.

4. **Sensors:** The ST16CF54A appears to have a number of sensors to detect intrusion attacks (One example is shown in Fig. 6)[2]. However, it appears that all

[2] This sensor is at the top metal layer (mesh), below the passivation. It is probably intended to detect physical intrusion, either through the encapsulation or through the passivation. We never encountered it application in the NagraVision / EchoStar implementation.

these sensors merely set software – accessible bits, and rely on the software to initiate protection measures in case these bits are set.
Such sensor implementation would not provide any protection against a "latched instruction" attack, since the code is not actually running. It may also indicate a potential vulnerability to "glitch" type attacks.

NOTE: Quite often software developers fail to utilize available sensor information. It appears that the sample ST16CF54A's software (NagraVision) does not use some of these sensors, at least during the card's initialization sequence (up to the end of ATR - Answer To Reset). For example, the NagraVision ST16CF54A works properly after decapsulation, it may be excited at frequencies down to 250 kHz, and powered to 8 V.
As a result, it is easy to acquire bus traffic during ATR - a significant, repeatable software segment. These data helps in understanding bus traffic acquired in a "latched instruction" attack.

5. **Modulo Arithmetic Processor (MAP):** This element takes a large portion of "real estate" on the ST16CF54A. Unlike the microprocessor itself, its exact function is not in the public domain.
However, the MAP's principal function and algorithms are known [Ref. 4].
Several approaches are therefore possible for obtaining its exact function, so that a clone can be built:

- Since the full microprocessor code can be extracted, it is possible to build a simple test apparatus that will feed known constants to a ST16CF54A IC on a probing table. The resulting bus response to these inputs would be known, and the exact function determined.
  Please note that at this stage, the information available on the sensors (see 3.4 above) should be sufficient to enable their neutralization.
- A prospective hacker could conceivably obtain a small number of sample ST16CF54A IC's in a programmable state (Issuer's state). These IC's could be programmed to determine the exact function of the MAP.
- The microprocessor could conceivably be forced back to the "Issuer State", obviating the need to obtain such samples from the manufacturer.

1. Cloning

Cloning the ST16CF54A should be possible using one of several approaches – all keyed on knowledge of processor program, as described above:

- Using a standard SmartCard microprocessor with a Modulo Arithmetic unit, it should be possible to produce a functionally – equivalent card, with customized parameters (ID etc.) to fit a particular system (card / IRD pairing etc.)

- A faster processor (either card-mounted microprocessor or a PC with an appropriate interface) could be used to emulate the ST16CF54A function.
- Knowledge of the ST16CF54A application code is a good starting point for application of "Glitch" techniques – that is, momentary changing the microprocessor's behavior by application of abnormal power, reset etc. If a successful "Glitch" is introduces at the right moment (for example – right at the execution of a security check's branch instruction), the result of that check is disregarded. Thus, an existing ST16CF54A based card may be reprogrammed to change its authorizations, or otherwise overcome CA system features.

Appendix A – code extraction

The following examples of ST16CF54A software (NagraVision / EchoStar implementation) were extracted and interpreted. For opcode convention see [Ref. 5]. Please note that the third example contained an "undocumented opcode" 0x9E. Logical analysis reveals this instruction to be "Load Stack Pointer to accumulator".

NOTE: For convenience, memory locations within the first 512 (decimal) bytes are displayed relative to address 0x0000, identified as "RAM". A "Xref" (Code or Sub) comment indicates a reference to the marked location from another. The comment is automatically generated by the disassembler, and just the first one or two are shown due to space limitations.

Example 1 – Swap nibbles routine:

Input: byte in accumulator. Output: byte in RAM location 0x0034 (and also in the accumulator). X register used. Called from two locations (listed)

```
428E     ; S u b r o u t i n e SWAP
428E
428E     sub_428E:                    ; CODE XREF: 4256, sub_437B+8
428E ;
428E   B6 34        lda RAM+0x34
4290   97           tax
4291   54           lsrx
4292   54           lsrx
4293   54           lsrx
4294   54           lsrx
4295   BF 34        stx RAM+0x34
4297   48           asla
4298   48           asla
4299   48           asla
429A   48           asla
429B   BA 34        ora RAM+0x34
429D   B7 34        sta RAM+0x34
429F   81           rts
429F                             ; End of function sub_428E
```

Example 2 - Text:

EEPROM locations E052 and F052 contain the text "NipPer Is a buTt liCker!"

Putting aside the question of who Nipper is and what are his moral qualities, one should note that such texts are often used to store essential data. A more complete analysis will no doubt reveal that.

Example 3 – Write and execute command in RAM

This routine (at 0x5A37) contains a mechanism of writing commands to the RAM and then jumping there to execute the just - written command. It contains a parameter passing mechanism, and uses an "undocumented" opcode 0x9E, not mentioned in Reference [5]. Our analysis shows this opcode to be a "Load Stack Pointer to accumulator". (Please note that our disassembler can be modified to handle undocumented opcodes, but this has not been done yet)

The routine has a parameter "m" located in the memory location immediately following the calling to the routine. "n" resides in the EEPROM at location 0xE051. The vector "P" starts in the EEPROM at location 0xE071, and the vector "Addr" starts (in our example card) at 0xE07A (EEPROM).

The routine performs the following (pseudo code description):

Update the returning address of the routine to be one location after the parameter.

If n=0 return.

For (x = 1; x <11; x++) if P[x] = m goto Calculate_Address;

Return;

Calculate_Address:

Goto the location whose address resides in the two consecutive bytes Addr[2*x+ n] & Addr[2*x+n+1]

The routine is called from many locations, the first one being 0x4022.

```
5A37      ; S u b r o u t i n e NO_NAME
5A37
5A37                sub_5A37:           ; CODE XREF: 4022
5A37  3F 03         clr RAM+3
5A39  BF 2D         stx RAM+0x2D
5A3B  B7 2E         sta RAM+0x2E
```

```
5A3D 9E          .byte 0x9E ;        ; SP -> Acc
5A3E 97          tax
5A3F 5C          incx
5A40 F6          lda , x
5A41 B7 21       sta RAM+0x21
5A43 E6 01       lda 1, x1
5A45 B7 22       sta RAM+0x22
5A47 6C 01       inc 1, x1
5A49 26 01       bne loc_5A4C
5A4B 7C          inc , x
5A4C
5A4C             loc_5A4C:           ; CODE XREF: sub_5A37+12
5A4C A6 C6       lda #0xC6 ;
5A4E B7 20       sta RAM+0x20
5A50 A6 81       lda #0x81 ;
5A52 B7 23       sta RAM+0x23
5A54 BD 20       jsr RAM+0x20
5A56 CE E0 51    ldx byte_E051
5A59 27 08       beq loc_5A63
5A5B
5A5B             loc_5A5B:           ; CODE XREF: sub_5A37+2A
5A5B D1 E0 71    cmp 0x71, x2
5A5E 27 08       beq loc_5A68
5A60 5A          decx
5A61 26 F8       bne loc_5A5B
5A63
5A63     loc_5A63:           ; CODE XREF: sub_5A37+22
5A63 BE 2D       ldx RAM+0x2D
5A65 B6 2E       lda RAM+0x2E
5A67 81          rts
5A68     ;
5A68
5A68             loc_5A68:           ; CODE XREF: sub_5A37+27_j
```

```
5A68 58        aslx
5A69 9F        txa
5A6A CB E0 51  add byte_E051
5A6D 97        tax
5A6E D6 E0 70  lda 0x70, x2
5A71 B7 21     sta RAM+0x21
5A73 D6 E0 71  lda 0x71, x2
5A76 B7 22     sta RAM+0x22
5A78 A6 CC     lda #0xCC ; 'Ì'
5A7A B7 20     sta RAM+0x20
5A7C BE 2D     ldx RAM+0x2D
5A7E B6 2E     lda RAM+0x2E
5A80 BC 20     jmp RAM+0x20
5A80                       ; End of function sub_5A37
5A80
```

References

[1] Thomson-SGS Smartcard Products Data Briefing Databook, 2nd Edition, 1996.

[2] Keifer, David S.: Window Etch Procedure for Multi-Layer VLSI. In Microelectronic Failure Analysis Desk Reference (3rd edition), Lee, T. W. and Pabbisetty, S.V., editors. ASM International 1993

[3] Anderson, Ross and Kuhn, Markus: Tamper Resistance – a Cautionary Note. Proceedings of the 2nd Workshop on Electronic Commerce, Oakland, CA. Nov. 1996. Available on the Internet at http://www.cl.cam.uk/users/rja14/

[4] Montgomery, P: Modular Multiplication without Trial Division. Mathematics of Computation, 4/1985.

[5] Motorola Inc. M6805UM/AD3: M6805 HMOS / M146805 CMOS Family User's Manual, Third edition

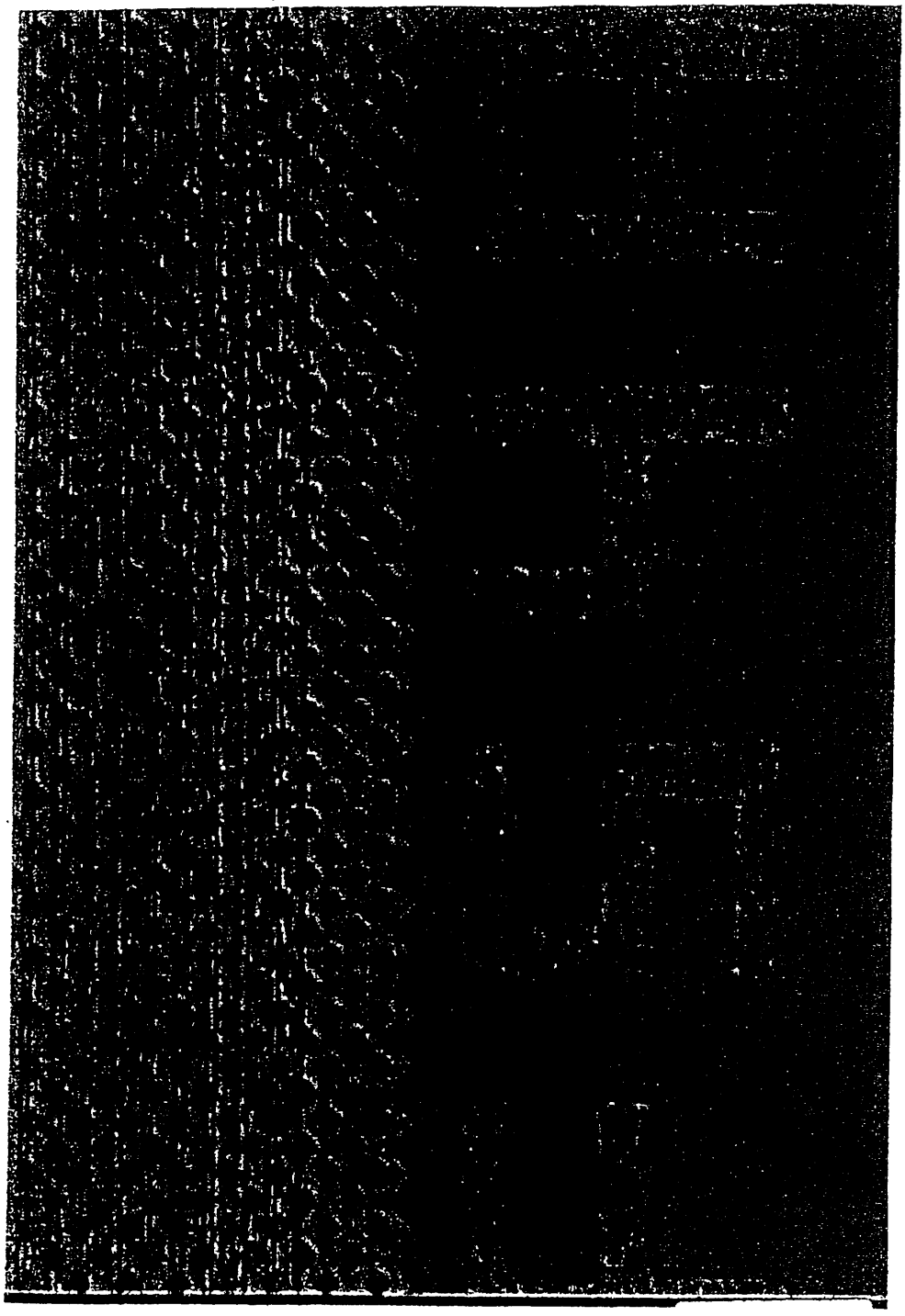

Fig. 1

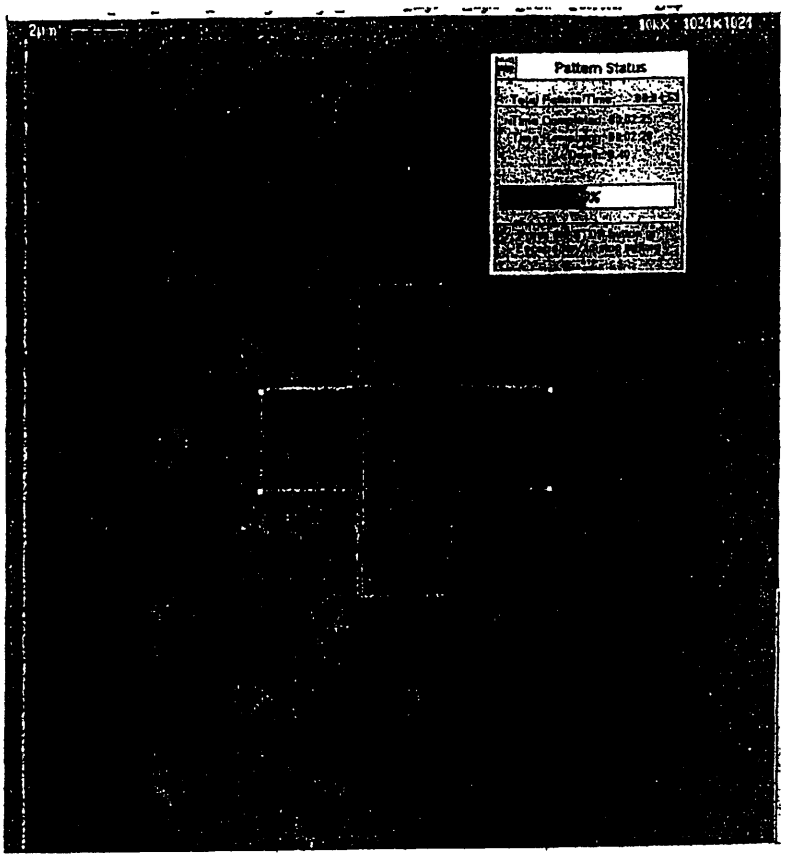

FIG. 2

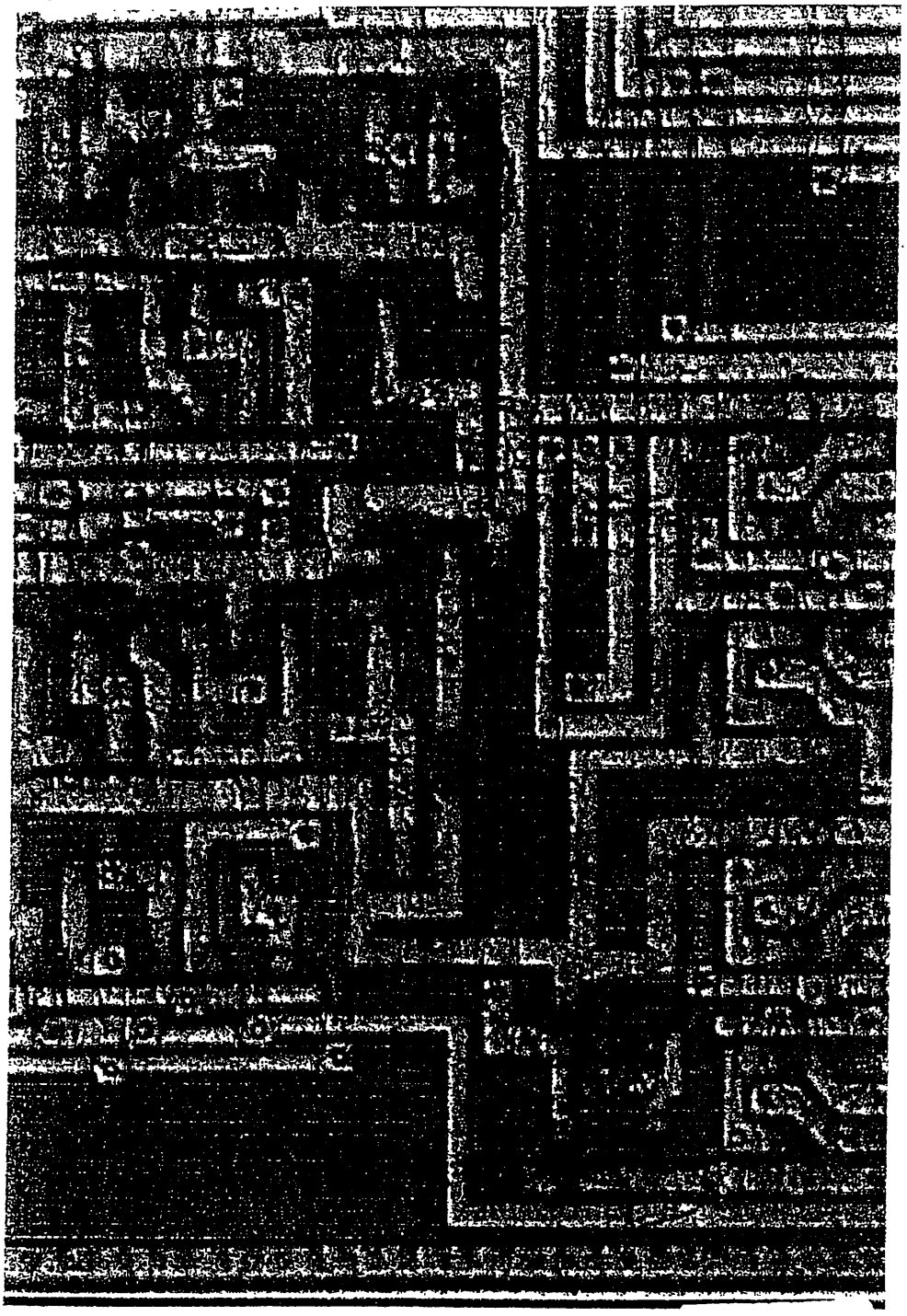

FIG. 3a

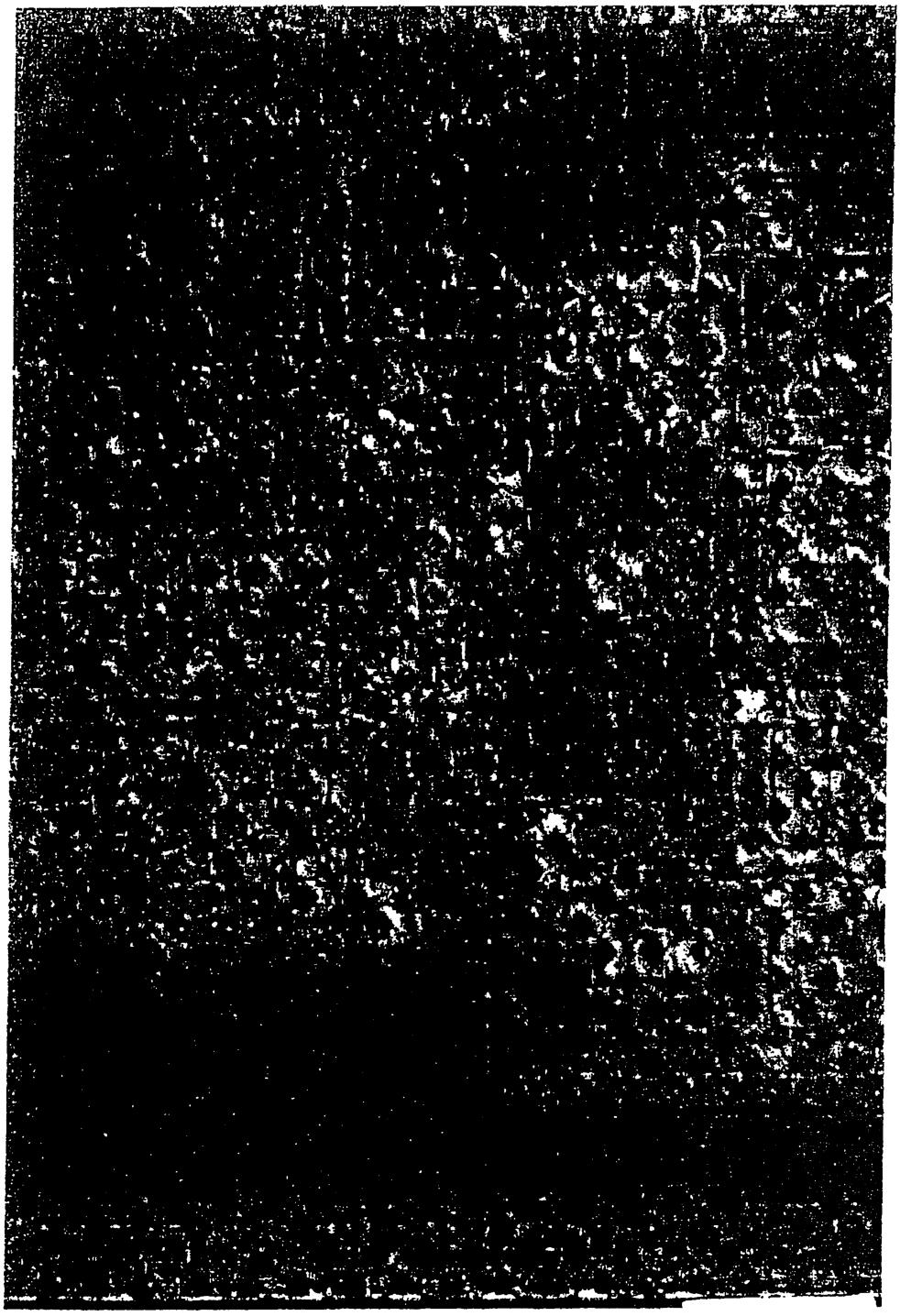

Fig 3

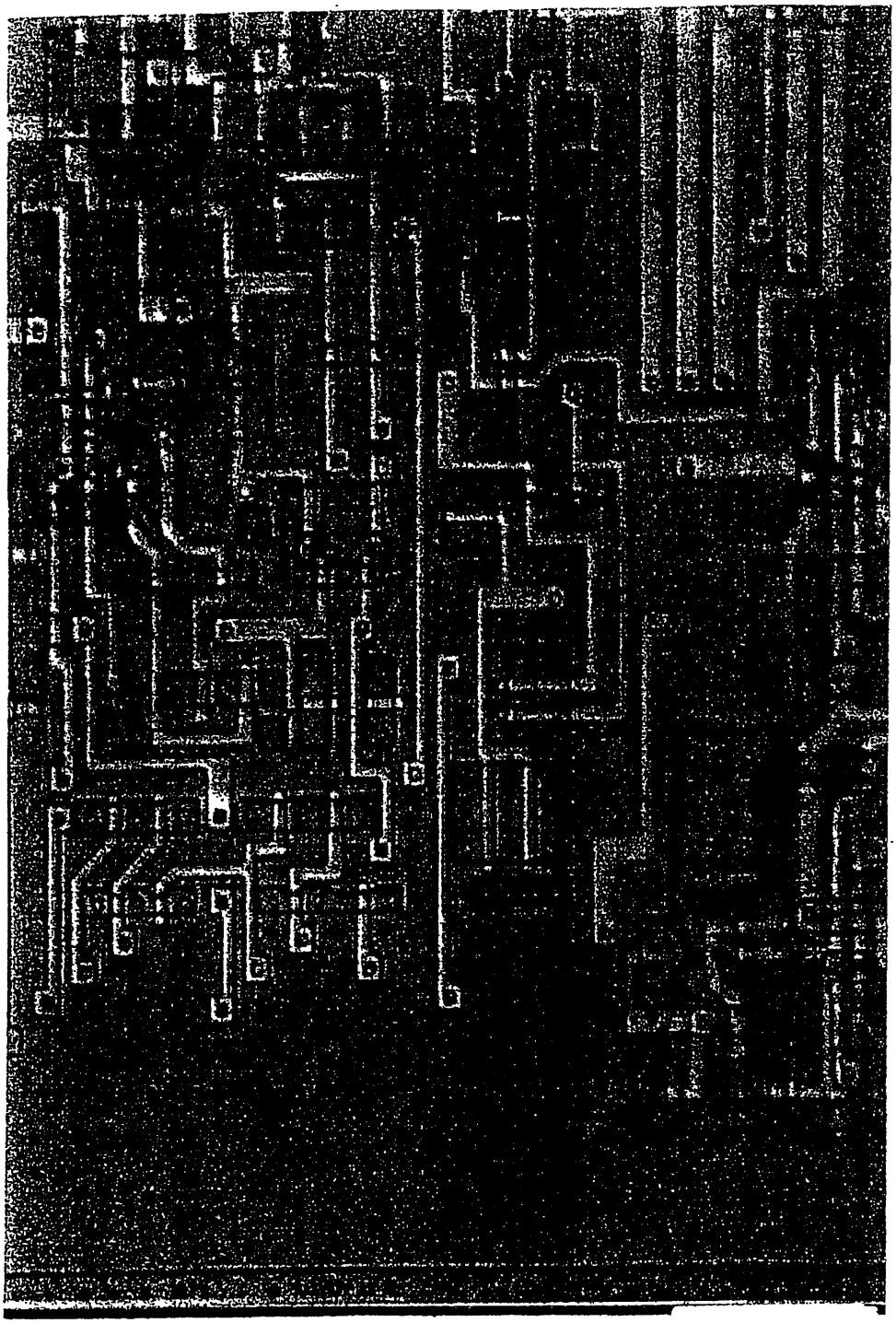

fig. 3c

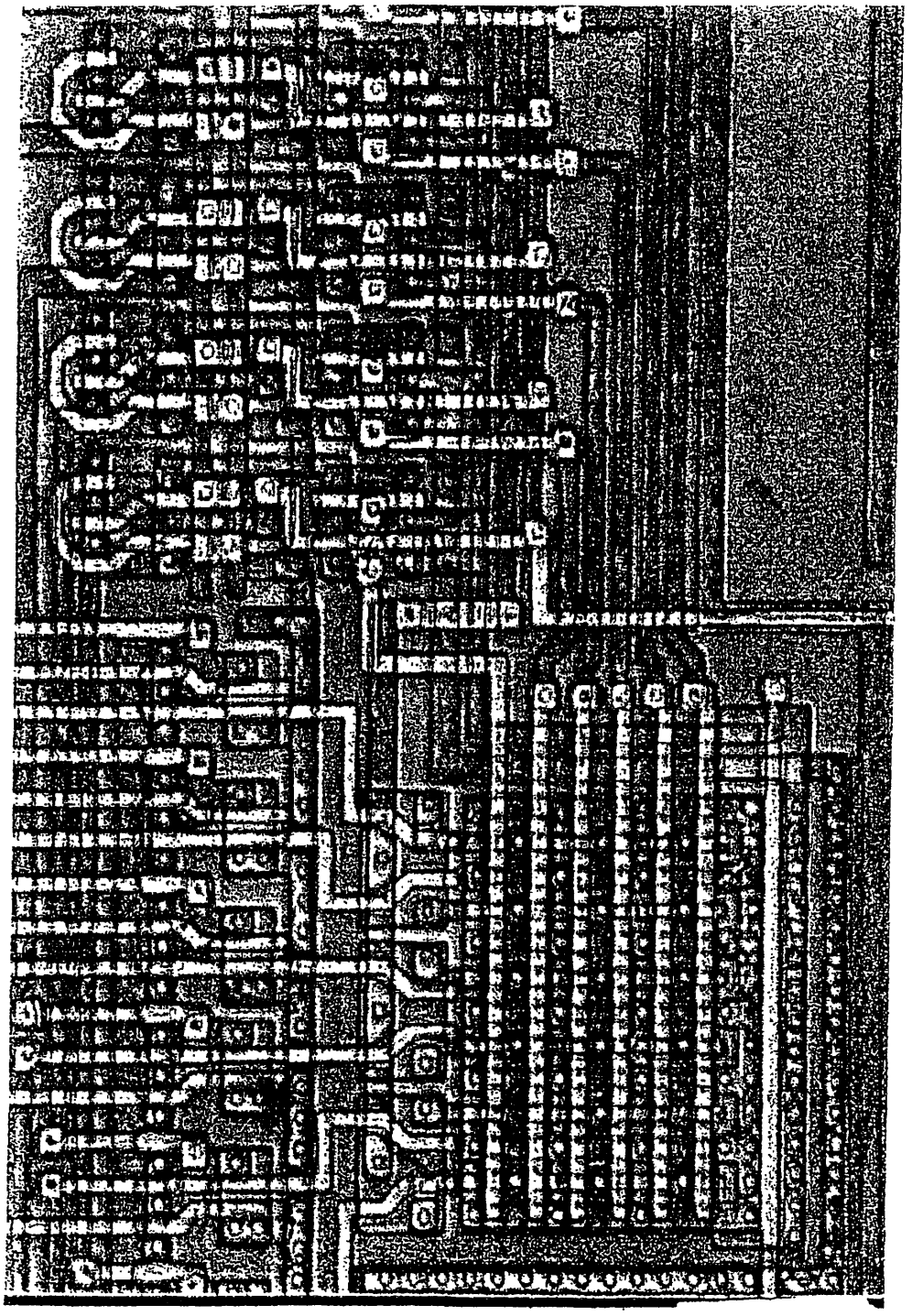

FIG. 4